[8] D. Bloor, K. Donnelly, P. J. Hands, P. Laughlin, and D. Lussey, "A metal-polymer composite with unusual properties," *J. Phys. D: Appl. Phys.*, vol. 38, no. 16, pp. 2851–2860, 2005.

[9] A. Balakrishnan, D. Kacher, A. Slocum, C. Kemper, and S. Warfield, "Smart retractor for use in image guided neurosurgery," in *Proc. Summer Bioeng. Conf.*, Key Biscayne, FL, 2003, pp. 895–896.

[10] K. Chinzei and K. Miler, "MRI guided surgical robot," in *Proc. Australian Conf. Robot. Autom.*, Sydney, Australia, 2001, pp. 50–55.

[11] D. Castro, L. Marques, U. Nunes, and A. de Almeida, "Tactile force control feedback in a parallel jaw gripper," in *Proc. IEEE Int. Symp. Ind. Electron.*, Guimarães, Portugal, 1997, pp. 884–888.

[12] H. Zhang and N. Chen, "Control of contact via tactile sensing," *IEEE Trans. Robot. Autom.*, vol. 16, no. 5, pp. 482–495, Oct. 2000.

[13] L. Birglen and C. Gosselin, "Fuzzy enhanced control of an underactuated finger using tactile and position sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 2320–2325.

[14] F. Vecchi, C. Freschi, S. Micera, A. Sabatini, and P. Dario, "Experimental evaluation of two commercial force sensors for applications in biomechanics and motor control," presented at the Int. Funct. Electr. Stimul. Soc., Aalborg, Denmark, 2000.

[15] M. Ferguson-Pell, S. Hagisawa, and D. Bain, "Evaluation of a sensor for low interface pressure applications," *Med. Eng. Phys.*, vol. 22, no. 9, pp. 657–663, 2000.

[16] K. Bachus, A. DeMarco, K. Judd, D. Horwitz, and D. Brodke, "Measuring contact area, force, and pressure for bioengineering applications: Using Fuji film and Tekscan systems," *Med. Eng. Phys.*, vol. 28, no. 5, pp. 483–488, 2006.

[17] J. Pavlovic, Y. Takahashi, J. Bechtold, R. Gustilo, and R. Kyle, "Can the Tekscan sensor accurately measure dynamic pressures in the knee joint?" in *Proc. Annu. Meeting Amer. Soc. Biomech.*, Clemson, SC, 1993, pp. 135–136.

[18] J. Otto, T. Brown, and J. Callaghan, "Static and dynamic response of a multiplexed-array piezoresistive contact sensor," *Exp. Mech.*, vol. 39, no. 4, pp. 317–323, 1999.

[19] R. Lakes, *Viscoelastic Solids*. Boca Raton, FL: CRC, 1999.

[20] E. Komi, J. Roberts, and S. Rothberg, "Evaluation of thin, flexible sensors for time-resolved grip force measurement," *IMechE Part C: J. Mech. Eng. Sci.*, vol. 221, pp. 1687–1699, 2007.

[21] (2009). [Online]. Available: http://www.tekscan.com/pdf/FlexiForce-Sensors-Manual.pdf

[22] (2011). [Online]. Available: http://www.interlinkelec.com/Force-Sensing-Resistor

[23] *Installation and Operations Manual for Stand-Alone F/T Sensor Systems*, ATI Industrial Automation, Inc., Apex, NC, 1997.

Keeping Multiple Moving Targets in the Field of View of a Mobile Camera

Nicholas R. Gans, Guoqiang Hu, Kaushik Nagarajan, and Warren E. Dixon

***Abstract*—This study introduces a novel visual servo controller that is designed to control the pose of the camera to keep multiple objects in the field of view (FOV) of a mobile camera. In contrast with other visual servo methods, the control objective is not formulated in terms of a goal pose or a goal image. Rather, a set of underdetermined task functions are developed to regulate the mean and variance of a set of image features. Regulating these task functions inhibits feature points from leaving the camera FOV. An additional task function is used to maintain a high level of motion perceptibility, which ensures that desired feature point velocities can be achieved. These task functions are mapped to camera velocity, which serves as the system input. A proof of stability is presented for tracking three or fewer targets. Experiments of tracking eight or more targets have verified the performance of the proposed method.**

***Index Terms*—Robust control, visual servoing, video surveillance.**

I. Introduction

Many vision-related control tasks cannot be formulated in terms of a specific goal pose or trajectory. Therefore, classical visual servoing methods (e.g., [1] and [2]) are not well suited to these problems. One task that is not well characterized by a goal pose or image is keeping multiple, moving objects in the camera field of view (FOV). Consider the scenario of crowd surveillance. A camera views the crowd and utilizes target segmentation and tracking methods to localize individuals of interest visible in the image. As the crowd moves and disperses, a controller must move and/or aim the camera in an attempt to keep all individuals in the FOV. Another scenario involves tracking several unmanned vehicles amid landmarks. Commands can be sent to the unmanned vehicles to track desired trajectories and avoid obstacles, but the vehicles must also be kept in the camera FOV to ascertain their pose in the workspace.

This paper presents a method to achieve the aforementioned tasks, extending our previous work in [3] and [4]. The method is rooted in classic image-based visual servoing [1], [2], [5], [6]; however, no goal image or goal feature trajectory is required. Rather than regulate error signals that are based on current and goal images, the proposed method regulates functions of the current image features. These task functions

Manuscript received February 9, 2011; accepted May 28, 2011. Date of publication June 30, 2011; date of current version August 10, 2011. This paper was recommended for publication by Associate Editor E. Marchand and Editor B. J. Nelson upon evaluation of the reviewers' comments. This work was supported in part by the Air Force Office of Scientific Research under Contract F49620-03-1-0381 and under Contract F49620-03-1-0170, in part by the Air Force Research Laboratory under Contract FA4819-05-D-0011, and in part by the U.S. Department of Energy under Grant DE-FG04-86NE37967. This work was accomplished as part of the DOE University Research Program in Robotics. This work was performed, in part, while N. Gans held a National Research Council Research Associateship Award at the Air Force Research Laboratory.

N. R. Gans and K. Nagarajan are with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: ngans@utdallas.edu; kxn094020@utdallas.edu).

G. Hu is with the Department of Mechanical and Nuclear Engineering, Kansas State University, Manhattan, KS 66506 USA (e-mail: gqhu@ksu.edu).

W. E. Dixon is with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: wdixon@ufl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2011.2158695

are suitable for task-priority kinematic control [7], [8]. The resulting controller allows feature points to move within the image, and the camera will move to deter feature points from leaving the FOV.

This study is partially inspired by the work of Bishop and Stilwell [9], [10], and Antonelli and Chiaverini [11]. These works applied task-priority kinematic control methods for redundant manipulators [7], [8], [12] to the field of mobile robot control. One approach in [11] controlled mean and variance of the vehicle coordinates. Similarly, it is possible to move the camera to control the distribution of targets in the image. If the mean of the target distribution is maintained near the image center, and the variance of feature coordinates is regulated to a suitably small value, objects will be inhibited from leaving the FOV. This paper also proves that mean regulation and variance regulation tasks do not interfere with each other; thus, the task-priority methods do not need to be considered for these two tasks. The control of our system also differs from previous task-priority controllers in that it includes robust control terms that ensure stability despite unknown disturbances.

We impose an additional task to maximize motion perceptibility (or simply, perceptibility) [13]. To the authors' knowledge, our works are the first to use perceptibility in the control feedback. The control of perceptibility for target tracking was also recently investigated in [14].

In [15]–[17], six moments of image segments are identified in goal and current images to build a full rank image Jacobian for position control. The concept of task functions and nullspace projection of multiple tasks has been used in visual servo control [6], [18]. In [18], Mansard and Chaumette extended the use of task functions to include a time-varying set of priorities that could be either deactivated or reprioritized. Our development uses task functions that have underdetermined Jacobians featuring nullspaces with large spans. The application of this study is also different, focusing on target tracking rather than positioning of a robot at a specific configuration. The problem of tracking three or fewer targets with a pair of pan/tilt/zoom cameras was presented in [19]. The approach in [19] exploits the geometry of cameras viewing points to define a desired relationship between the points in the two images.

II. Camera Model and Image Jacobian for a Camera Model

Consider a camera with coordinate frame $\mathcal{F}_c(t)$. The camera views a collection of k feature points in front of the camera. The coordinates of the feature points in the image are denoted as

$$m_i = [x_i, y_i]^T, \quad \forall i \in \{1 \ldots k\}. \tag{1}$$

Given the collection of k feature points, along with their coordinates and velocity vectors, a state position vector $m(t)$ and state velocity vector $\dot{m}(t)$ are defined as

$$m = [m_1^T, m_2^T, \ldots, m_k^T]^T, \quad \dot{m} = [\dot{m}_1^T, \dot{m}_2^T, \ldots, \dot{m}_k^T]^T.$$

If the feature points are static in an inertial frame, the feature point velocity in the image plane is given as a function of the camera velocity $v_c(t) \in \mathbb{R}^6$ by the relationship

$$\dot{m} = L v_c \tag{2}$$

where $L(t) \in \mathbb{R}^{2k \times 6}$ is the image Jacobian. The image Jacobian for the feature points is constructed by concatenating a set of k submatrices $L_i(t) \in \mathbb{R}^{2\times 6}$ [6], with $L_i(t)$ given as

$$L_i = \begin{bmatrix} \frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & -x_i y_i & 1 + x_i & -y_i \\ 0 & \frac{1}{Z_i} & \frac{y_i}{Z_i} & -1 - y_i^2 & x_i y_i & x_i \end{bmatrix} \tag{3}$$

where Z_i is the depth of point i in $\mathcal{F}_c(t)$.

In the case that the feature points are not static in an inertial frame, the time derivative of the feature points is given by

$$\dot{m} = L v_c + \varepsilon \tag{4}$$

where $\varepsilon(t)$ is a (generally unknown) vector that reflects the apparent motion of the points in the image because of the 3-D motion of the targets.

There are several reasonable assumptions that are required for the following development.

Assumption 1: The system is initially well conditioned, such that $\forall i$, $Z_i(0) > c_z$, where c_z is some positive constant, and m_i are within the camera FOV.

Assumption 2: The velocity of all targets is bounded by a known constant c_p.

Assumption 3: The 3-D coordinates of all targets points remain finite.

Assumption 4: The distance between targets is bounded from below by a known constant (i.e., the targets cannot intersect each other).

Assumption 5: The depth of all points $Z_i(t)$ is known.[1]

If $Z_i(0) > c_z > 0$, $\forall i$, then subsequent analysis will show that $Z_i(t) > 0$, $\forall i, t$ (see Theorem 2). Given $Z_i(t) > 0$, $\forall i, t$, and Assumptions 1 and 2, it can be shown that $\varepsilon(t)$ is bounded such that $\|\varepsilon(t)\| < c_\varepsilon \; \forall t$, where $c_\varepsilon \in \mathbb{R}^+$ is a known constant. $L(t) \in \mathcal{L}_\infty$ if $x_i(t)$, $y_i(t) \in \mathcal{L}_\infty \, \forall i$, and $Z_i(t) > 0 \, \forall \, i, t$. Specifically, $\|L(t)\| < c_L$, where $c_L \in \mathbb{R}^+$ is a known constant. Therefore, under Assumption 1, $L(0)$ is bounded. The subsequent development ensures that $L(t)$ is bounded for all t, i.e., $\|L(t)\| < c_L \; \forall t$.

III. Task Function-Based Camera Control

The control objective is to keep a set of feature points, which represent multiple moving targets, within the camera FOV. Motivated by this objective, a set of task functions are defined by using image feature coordinates. By regulating the task functions, feature points can be inhibited from leaving the FOV. To avoid competition between task functions, task-priority kinematic control [7], [8] is used.

Let $\phi(t) \in \mathbb{R}^l$ denote a task function of the feature point coordinates $m_i(t), i \in \{1 \ldots k\}$, as

$$\phi = f(m_1, \ldots, m_k)$$

with derivative

$$\dot{\phi} = \sum_{i=1}^{k} \frac{\partial f}{\partial m_i} \dot{m}_i = J(m)\dot{m} \tag{5}$$

where $J(m) \in \mathbb{R}^{l \times 2k}$ is the task Jacobian matrix. The task functions that are developed subsequently are of dimension $l \leq 2$.

The task is to drive the feature points along a desired velocity $\dot{m}_\phi(t) \in \mathbb{R}^{2k}$ such that $\phi(t)$ follows a desired trajectory $\phi_d(t)$. Given the underdetermined structure of the Jacobian matrix, there are infinite solutions to this problem. The typical solution is to use the Moore–Penrose pseudoinverse [8] as

$$\begin{aligned} \dot{m}_\phi &= J^\dagger \left[\dot{\phi}_d - \gamma(\phi - \phi_d)\right] \\ &= J^T \left(J J^T\right)^{-1} \left[\dot{\phi}_d - \gamma(\phi - \phi_d)\right] \end{aligned} \tag{6}$$

[1] Sensitivity of visual servoing to depth estimation errors is a well-known issue, but it is not the focus of this paper. See [20] and references therein for discussion of estimating point depths in visual servoing. Stability proofs will assume the ideal case that the image Jacobian is known, and experimental results demonstrate the system performance when point depths are roughly approximated based on target size in the image.

where γ is a positive scalar gain constant, and $J^{\dagger}(m) \in \mathbb{R}^{2k \times l}$ denotes the right pseudoinverse of $J(m)$. Based on (6), the camera velocity $v_c\,(t)$ is designed as

$$v_c = L^{+}\dot{m}_{\phi} = \left(L^T L\right)^{-1} L^T \dot{m}_{\phi} \tag{7}$$

where $L^{+}\,(t) \in \mathbb{R}^{6 \times 2k}$ denotes the left pseudoinverse of $L\,(t)$.[2] In the case of three feature points, L is square, and the general inverse L^{+} is replaced with the inverse L^{-1}. The potential singularity of a square L is a well-known problem, and a damped general inverse can be used to maintain a smooth trajectory near a singularity [8], [21].

Task functions can be combined in numerous ways [8]. Consider two tasks $\phi_a\,(t)$ and $\phi_b\,(t)$ with associated task Jacobian matrices $J_a\,(m_a)$ and $J_b\,(m_b)$ and resulting desired feature velocities $\dot{m}_a\,(t)$ and $\dot{m}_b\,(t)$. One method is to choose one task as primary, and project other tasks into the nullspace of the primary task derivative [7], [8] as

$$v_c = L^{+}\left(J_a^{\dagger}\dot{\phi}_a + (I - J_a^{\dagger}J_a)J_b^{\dagger}\dot{\phi}_b\right). \tag{8}$$

The approach in (8) will prevent the velocity vectors from competing and negating each other, as the primary task will always be accomplished. Lower priority control tasks will be achieved if they do not oppose higher priority tasks. Tertiary, quaternary, etc., tasks can be prioritized by repeating this process and projecting each subsequent task into the nullspace of the preceding task Jacobians.

IV. Control Development

Three task functions are used in this controller. Two task functions are designed to regulate the mean and variance of the feature point coordinates. Regulating the mean at the camera center inhibits the feature points from leaving center of the FOV. Regulating the variance will restrict the distance between the feature points and the mean and work to keep feature points away from the edge of the FOV. A third task function maximizes motion perceptibility.

If variance is successfully regulated to a suitably small goal value, a subset of feature points can be guaranteed to stay in view. Given a scalar p and a random variable x with mean $\bar{x}$ and variance σ_x^2, Chebyshev's inequality states [22]

$$P(|x - \bar{x}| \geq p\sigma_x) \leq \frac{1}{p^2}.$$

This provides a bound on the number of elements in a sample that are farther than a specified distance of the mean. Specifically, Chebyshev's inequality proves that at least 75% of all values are within two standard deviations of the mean, and at least 89% of values are within three standard deviations. Consider a camera with a 512×512 pixel FOV and uniformly distributed feature points in the image. For arbitrary distribution of feature points, regulating the variance to 86^2 will ensure that at least 89% of all points are in the FOV.

A. Task Function for the Mean of the Image Points

Let $\phi_m\,(t) \in \mathbb{R}^2$ denote a task function that is defined as the sample mean

$$\phi_m = \frac{1}{k}\sum_{i=1}^{k} m_i = \bar{m}.$$

The time derivative of $\phi_m\,(t)$ is given by

$$\begin{aligned}\dot{\phi}_m &= \frac{1}{k}\sum_{i=1}^{k}\frac{\partial\phi_m}{\partial m_i}\dot{m}_i = J_m\dot{m}\\ &= J_m\,(Lv_c + \varepsilon)\end{aligned} \tag{9}$$

where $J_m\,(t) \in R^{2 \times 2k}$ is a task function Jacobian that is defined as

$$J_m = \frac{1}{k}\left[I_2, \ldots, I_2\right] \tag{10}$$

where I_2 is the 2×2 identity matrix and is repeated k times. It can be determined that $\|J_m\|_2 = 1/\sqrt{k}$.

Let $\phi_{m\,d}\,(t)$ denote a desired task function trajectory with a known derivative $\dot{\phi}_{m\,d}\,(t)$. A robust feedback control can be used to generate a feature point velocity that will track the desired value of $\phi_{m\,d}\,(t)$. This feature point velocity is denoted as $\dot{m}_m\,(t) \in \mathbb{R}^{2k}$, and is given by

$$\dot{m}_m = -J_m^{\dagger}\left(\gamma_m\tilde{\phi}_m - \dot{\phi}_{m\,d} + \frac{c_{\varepsilon}^2}{k\kappa_m}\tilde{\phi}_m\right) \tag{11}$$

where $\tilde{\phi}_m(t) = \phi_m(t) - \phi_{m\,d}(t)$, and $\gamma_m, \kappa_m \in \mathbb{R}^{+}$ are constant gains. The final term in (11) is a robust disturbance–rejection term, which is designed based on the stability analysis in Section IV-E.

B. Task Function for the Variance of the Image Points

A task function $\phi_v\,(t) \in \mathbb{R}^2$ is given by the sample variance of the feature points in the current image

$$\phi_v = \frac{1}{k}\sum_{i=1}^{k}\begin{bmatrix}(x_i - \bar{x})^2\\ (y_i - \bar{y})^2\end{bmatrix}$$

where $\bar{x}\,(t)$ and $\bar{y}\,(t)$ are the sample mean (in the current image) of the x and y components of $m_i\,(t)\,, i \in \{1 \ldots k\}$.

The time derivative of $\phi_v\,(t)$ is given by

$$\dot{\phi}_v = J_v\dot{m} = J_v\,(Lv_c + \varepsilon) \tag{12}$$

where $L(t)$, $v_c(t)$, and $\varepsilon(t)$ were given in (4), and $J_v\,(t) \in \mathbb{R}^{2 \times 2k}$ is a task function Jacobian that is given by

$$J_v = \frac{2}{k}\left[\begin{matrix}x_1 - \bar{x} & 0\\ 0 & y_1 - \bar{y}\end{matrix}, \begin{matrix}x_2 - \bar{x} & 0\\ 0 & y_2 - \bar{y}\end{matrix}, \ldots, \begin{matrix}x_k - \bar{x} & 0\\ 0 & y_k - \bar{y}\end{matrix}\right]. \tag{13}$$

The derivation of J_v can be seen in [3] and [4], and J_v is the same as the Jacobian in [11] up to a constant scale factor. J_v is singular if all image points are collinear. The developments of this paper assume that J_v remains full rank. The subsequent stable controller will prevent some collinear configurations of points, and the damped least-squares general inverse can be used to avoid singularities as well [8], [21].

By the definition of sample mean, $x_i(t)$, $y_i(t) \in \mathcal{L}_\infty\,\forall i$ implies $\bar{x}$, $\bar{y} \in \mathcal{L}_\infty$. From (13), $J_v(t) \in \mathcal{L}_\infty$ iff $x_i(t)$, $y_i(t)$, $\bar{x}$, $\bar{y} \in \mathcal{L}_\infty\,\forall i$. The subsequent development ensures that if x_i, and y_i are in the image at initial time 0, all terms x_i, and y_i are bounded (see Theorem 2). Therefore, $\|J_v\| \leq c_v$, where c_v is a known, constant bound that is determined from the maximum image coordinates.

To regulate the variance to a desired trajectory $\phi_{v\,d}\,(t)$ with a known, smooth derivative $\dot{\phi}_{v\,d}\,(t)$, the feature point velocity $\dot{m}_v\,(t) \in \mathbb{R}^{2k}$ can be designed as

$$\dot{m}_v = -J_v^{\dagger}\left(\gamma_v\tilde{\phi}_v - \dot{\phi}_{v\,d} + \frac{c_v^2 c_{\varepsilon}^2}{\kappa_v}\tilde{\phi}_v\right). \tag{14}$$

[2] Note that $J\,(m)$ is underdetermined (i.e., more columns than rows), and $L(m)$ is overdetermined (i.e., more rows than columns). Therefore, the Moore–Penrose pseudoinverse is constructed differently for each matrix and care is taken to denote them differently. Specifically, † denotes the right inverse used for $J\,(t)$, and $+$ denotes the left inverse used for $L\,(t)$. During implementation, the psuedoinverse for both overdetermined and underdetermined matrices can be calculated numerically using singular value decomposition. This offers some numerical benefits during operation.

where $\tilde{\phi}_v(t) = \phi_v(t) - \phi_{vd}(t)$, $\gamma_v \in \mathbb{R}^+$ is a constant gain, and $\kappa_v \in \mathbb{R}^+$ is a constant, robust, disturbance–rejection gain.

We show that the tasks of regulating mean and variance will not interfere with each other, and $\dot{m}_m + \dot{m}_v$ can be used in place of the nullspace projection of (8). This result has potential impact on other control routines that use mean and variance [11], [23].

Theorem 1: Given the Jacobian matrices J_m and J_v

$$J_m J_v^\dagger = J_v J_m^\dagger = 0.$$

Proof: The pseudoinverse $J_v^\dagger(t) \in \mathbb{R}^{2k\times 2}$ is given in closed form as

$$J_v^\dagger = \frac{k}{2}\begin{bmatrix} \frac{x_1-\bar{x}}{\sum_{i=1}^k (x_i-\bar{x})^2} & 0 \\ 0 & \frac{y_1-\bar{y}}{\sum_{i=1}^k (y_i-\bar{y})^2} \\ \vdots & \vdots \\ \frac{x_k-\bar{x}}{\sum_{i=1}^k (x_i-\bar{x})^2} & 0 \\ 0 & \frac{y_k-\bar{y}}{\sum_{i=1}^k (y_i-\bar{y})^2} \end{bmatrix}^T . \tag{15}$$

Using (10) and (15)

$$J_m J_v^\dagger = \frac{1}{2}\begin{bmatrix} \frac{k\bar{x}-k\bar{x}}{\sum_{i=1}^k (x_i-\bar{x})^2} & 0 \\ 0 & \frac{k\bar{y}-k\bar{y}}{\sum_{i=1}^k (y_i-\bar{y})^2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

It can similarly be shown that $J_v J_m^\dagger = 0$. ■

Combining (8), (11), (14), and the result of Theorem 1, the resulting feature point velocity for prioritizing the tasks is

$$\begin{aligned} \dot{m} &= \dot{m}_m + (I - J_m^\dagger J_m)\dot{m}_v \\ &= J_m^\dagger \dot{\phi}_m + (I - J_m^\dagger J_m) J_v^\dagger \dot{\phi}_v \\ &= J_m^\dagger \dot{\phi}_m + J_v^\dagger \dot{\phi}_v = \dot{m}_m + \dot{m}_v . \end{aligned}$$

C. *Task Function for Perceptibility of Image Points*

Sharma and Hutchinson presented the concept of motion perceptibility in [13]. Related to the concept of manipulability [24], perceptibility gives a measure of how well a camera can perceive the motion of objects in the FOV. Roughly speaking, if perceptibility is high, small object or camera velocities will result in notable feature velocities in the image plane (e.g., high optical flow). This is especially important if there are more than three feature points, as the available feature point velocities are constrained because of an overdetermined image Jacobian. Maintaining a high perceptibility helps to ensure a larger span of available feature point velocity vectors.

Perceptibility is a scalar function of the image Jacobian $L(t)$ that is defined as

$$w_v = \sqrt{\det(L^T L)} = \prod_{i=1}^{6} s_i$$

where $s_i(t) \in \mathbb{R}^+$ are the nonzero singular values of $L(t)$. Maximization of $w_v(t)$ is accomplished by maximization of each $s_i(t)$. The matrix $L^T(t)L(t) \in \mathbb{R}^{6\times 6}$ is symmetric and positive definite (SPD). Hardamard's inequality states that the determinant of an SPD matrix is less than or equal to the product of its diagonal elements [25]. Thus, increasing the product of diagonals will increase the upper bound on w_v, potentially improving perceptibility.

The product of singular values of $L^T(t)L(t)$ is given by

$$\begin{aligned} p = &\left(\sum_{j=1}^{k} 1/Z_j^2\right)^2 \left(\sum_{j=1}^{k} (x_j^2 + y_j^2)/Z_j^2\right) \left(\sum_{j=1}^{k} (x_j^2 + y_j^2)/Z_j^2\right) \\ &\times \left(\sum_{j=1}^{k} x_j^2 y_j^2 + (y_j^2+1)^2\right) \left(\sum_{j=1}^{k} x_j^2 y_j^2 + (x_j^2+1)^2\right). \end{aligned} \tag{16}$$

Based on (16), increasing $x_i^2 + y_i^2$ will increase the perceptibility. A task function for perceptibility is selected as

$$\phi_p = \frac{1}{\sum_{i=1}^{k} (x_i^2 + y_i^2)}.$$

Since it is desired to increase $p(L^T L)$, regulating $\phi_p(t)$ to 0 will increase the trace. The time derivative of $\phi_p(t)$ is

$$\begin{aligned} \dot{\phi}_p &= -2\phi_p^2 \sum_{i=1}^{k} \begin{bmatrix} x_i & y_i \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} \\ &= J_p(m)\dot{m} = J_p(m)(L v_c + \varepsilon) \end{aligned}$$

where $J_p(m) \in \mathbb{R}^{1\times 2k}$ is the task function Jacobian for perceptibility and is undefined only for the nongeneral case that $\forall i,\ m_i = [0,\quad 0]^T$.

To regulate $\phi_p(t)$ to 0, the feature point velocity $\dot{m}_p(t) \in \mathbb{R}^{2k}$ is designed as

$$\dot{m}_p = -\gamma_p J_p^\dagger \phi_p \tag{17}$$

where γ_p is a positive scalar gain constant. A robust feedback term is not used for perceptibility regulation. This is because of the fact that $\phi_p(t)$ will be a low-priority task and, thus, blocked by higher priority tasks and unlikely to ever become zero. The extra control effort and complexity that result from a robust control term is not deemed essential.

D. *Cascaded Camera Control Law*

As shown in Theorem 1, regulating the mean and variance will not conflict; therefore, they are chosen as the primary tasks to keep the feature points centered in the FOV and to restrict the distance between the feature points and the image center. High perceptibility will allow these tasks to work more efficiently by ensuring that larger available feature velocities are available. For this reason, increasing perceptibility is chosen as the lower priority task, and it cannot interfere with the regulation of mean or variance. The designed image feature velocities that are given in (11), (14), and (17) are used in the nullspace projection camera velocity (8) to yield the overall controller

$$\begin{aligned} v_c &= L^+ \left(\dot{m}_m + \dot{m}_v + \left(I - J_m^\dagger J_m\right)\left(I - J_v^\dagger J_v\right)\dot{m}_p\right) \\ &= L^+ \left(\dot{m}_m + \dot{m}_v + \left(I - J_m^\dagger J_m - J_v^\dagger J_v\right)\dot{m}_p\right) \end{aligned} \tag{18}$$

where the independence of $\dot{m}_m$ and $\dot{m}_v$ that is proved in Theorem 1 has been exploited.

In the subsequent stability analysis, if $L(t),\ J_v(t) \in \mathcal{L}_\infty\ \forall t$, the controller will stabilize the mean and variance. This implies that $\phi_m(t),\ \phi_v(t) \in \mathcal{L}_\infty$, which implies $x_i(t),\ y_i(t) \in \mathcal{L}_\infty\ \forall i$. In practice, $x_i(t),\ y_i(t) \in \mathcal{L}_\infty\ \forall i$ does not imply that each feature remains in the FOV. The tracking controller will inhibit features from leaving the FOV, but a feature may escape. In these cases, there are several alternatives that preserve all boundedness properties. The lost feature can be dropped from consideration, and all matrices and vectors are reformulated from this time with one fewer feature point. Alternatively, the lost point can be kept constant at its last known value, which will cause the tracker to attempt to bring it back in view. These two methods can be merged by keeping the lost point at its last known value for a finite time and then scaling the lost point with a time-varying, decreasing gain, similar to the approach in [26].

E. Stability Analysis for Three or Fewer Points

If there are three or fewer feature points in view of the camera, the camera velocities that are given in (11), (14), (17), and (18) will regulate the mean and variance to within an arbitrarily small distance from the desired values.

Theorem 2: Given $k \leq 3$ feature points, if the matrix bounds $\|L\| < c_L$ and $\|J_v\| < c_v$ exist for known, positive scalars c_L and c_v, then the feedback control law that is given by (11), (14), (17), and (18) will ensure that the mean and variance tracking errors are uniformly ultimately bounded such that

$$|\tilde{\phi}_m|, |\tilde{\phi}_v| \leq \sqrt{\beta_0 e^{-\gamma t} + \frac{K}{\gamma}} \tag{19}$$

where β_0 is some positive, scalar constant, $\gamma = \min\{\gamma_m, \gamma_v\}$, and K is a scalar constant that can be made arbitrarily small.

Proof: For $k = 3$ points, L is the square and $LL^{-1} = I$. For $k \leq 2$, L is underdetermined; therefore, the left pseudoinverse is used, and $LL^{\dagger} = I$. For all k, $J_m J_m^{\dagger} = I$. For $k = 1$, there is no variance, and all variance terms can be dropped from the control. For $k \geq 2$, $J_v J_v^{\dagger} = I$. Combining (9), (11), (14), (17), and (18), and using the fact that $J_m J_v^{\dagger} = J_v J_m^{\dagger} = 0$, the closed-loop time derivative of $\phi_m(t)$ is

$$\begin{aligned} \dot{\phi}_m &= J_m \dot{m} = J_m (L v_c + \varepsilon) \\ &= -\gamma_m \tilde{\phi}_m + \dot{\phi}_{md} - \frac{c_\varepsilon^2}{k\kappa_m} \tilde{\phi}_m + J_m \varepsilon. \end{aligned} \tag{20}$$

Similarly, combining (11), (12), (14), (17), and (18), the closed-loop time derivative of $\phi_v(t)$ can be found as

$$\dot{\phi}_v = -\gamma_v \tilde{\phi}_v + \dot{\phi}_{vd} - \frac{c_v^2 c_\varepsilon^2}{\kappa_v} \tilde{\phi}_v + J_v \varepsilon. \tag{21}$$

Define a candidate Lyapunov function

$$V = \frac{1}{2} \left\|\tilde{\phi}_m\right\|^2 + \frac{1}{2} \left\|\tilde{\phi}_v\right\|^2 \tag{22}$$

with time derivative

$$\begin{aligned} \dot{V} &= \tilde{\phi}_m^T \dot{\tilde{\phi}}_m + \tilde{\phi}_v^T \dot{\tilde{\phi}}_v \\ &= \tilde{\phi}_m^T \left(\dot{\phi}_m - \dot{\phi}_{md}\right) + \tilde{\phi}_v^T \left(\dot{\phi}_v - \dot{\phi}_{vd}\right). \end{aligned} \tag{23}$$

The substitution of (20) and (21) gives

$$\begin{aligned} \dot{V} &= \tilde{\phi}_m^T \left(-\gamma_m \tilde{\phi}_m - \frac{c_\varepsilon^2}{k\kappa_m} \tilde{\phi}_m + J_m \varepsilon\right) \\ &\quad + \tilde{\phi}_v^T \left(-\gamma_v \tilde{\phi}_v - \frac{c_v^2 c_\varepsilon^2}{\kappa_v} \tilde{\phi}_v + J_v \varepsilon\right) \\ &\leq -\gamma_m \left\|\tilde{\phi}_m\right\|^2 + \left\|\tilde{\phi}_m\right\| \|J_m\| \|\varepsilon\| - \frac{c_\varepsilon^2}{k\kappa_m} \left\|\tilde{\phi}_m\right\|^2 \\ &\quad - \gamma_v \left\|\tilde{\phi}_v\right\|^2 + \left\|\tilde{\phi}_v\right\| \|J_v\| \|\varepsilon\| - \frac{c_v^2 c_\varepsilon^2}{\kappa_m} \left\|\tilde{\phi}_v\right\|^2. \end{aligned} \tag{24}$$

Completing the square on (24) and using the known bounds on $\|J_m\|$, $\|J_v\|$ and ε gives

$$\begin{aligned} \dot{V} &\leq -\gamma_m \left\|\tilde{\phi}_m\right\|^2 - \frac{c_\varepsilon^2}{k\kappa_m} \left[\left(\left\|\tilde{\phi}_m\right\| - \frac{\sqrt{k}\kappa_m}{2c_\varepsilon}\right)^2 - \frac{k\kappa_m^2}{4c_\varepsilon^2}\right] \\ &\quad - \gamma_v \left\|\tilde{\phi}_v\right\|^2 - \frac{c_v^2 c_\varepsilon^2}{\kappa_v} \left[\left(\left\|\tilde{\phi}_v\right\| - \frac{\kappa_v}{2c_v c_\varepsilon}\right)^2 - \frac{c_v^2 \kappa_v}{4c_\varepsilon^2}\right] \\ &\leq -\gamma_v \left\|\tilde{\phi}_v\right\|^2 - \gamma_m \left\|\tilde{\phi}_m\right\|^2 + \frac{\kappa_m + \kappa_v}{4}. \end{aligned} \tag{25}$$

Therefore, using (22), (25) can be rewritten as

$$\dot{V} \leq -2\gamma V + K \tag{26}$$

where $K = (\kappa_m + \kappa_v)/4$ can be made arbitrarily small by the choice of the positive robust control terms κ_m and κ_v. The solution of the differential inequality (26) can be found as

$$V \leq e^{-2\gamma t} V(0) + \frac{K}{2\gamma} \left(1 - e^{-2\gamma t}\right). \tag{27}$$

Defining $\Phi(t) = \left[\tilde{\phi}_m, \tilde{\phi}_v\right]^T$, (27) can be rewritten as

$$\|\Phi\| \leq \sqrt{e^{-2\gamma t} \|\Phi(0)\|^2 + \frac{K}{2\gamma} (1 - e^{-2\gamma t})}.$$

■

Note that c_ε and c_v need to be known for the control law, but they do not affect the ultimate bound of the error.

V. Experimental Results

Experiments were performed to examine the performance of controlling mean and variance to track multiple moving targets. Perceptibility was not included in these experiments. The experiments used a pioneer 3-DX mobile robot. As inputs, the robot has a single linear velocity along the z-axis of $\mathcal{F}_c$ and a single angular velocity about the y-axis. This is accommodated in the proposed controller by the usage of the third and fifth columns of the image Jacobian L that is defined in (2). A laptop with an Intel Core Duo 2.667 GHz processor performed all image processing and high-level control calculations. The camera is a Matrix Vision BlueFox, with a resolution of 1024×768 pixels. The targets are two circuit boards with IR LEDs mounted on the surface. One of the targets has five LEDs, and the other has three LEDs. The centroid of each LED gives the feature point m_i. This gives eight feature points to be kept in the FOV. The frame rate of the camera when including all image processing is approximately 24 frames/s.

The targets were continuously moved in different directions around the robot. The distance between the targets was varied, as was the distance of the targets to the robot. The robot successfully keeps the two targets in its FOV. Figs. 1 and 2 show the error signals and the velocities over time. There is no convergence in the graphed signals because the targets were in motion throughout experiment. However, it is clear that the error and velocity signals are bounded for the duration of the experiment. While the results shown are for 260 s, experiments have been conducted for several minutes without the robot losing sight of the targets.

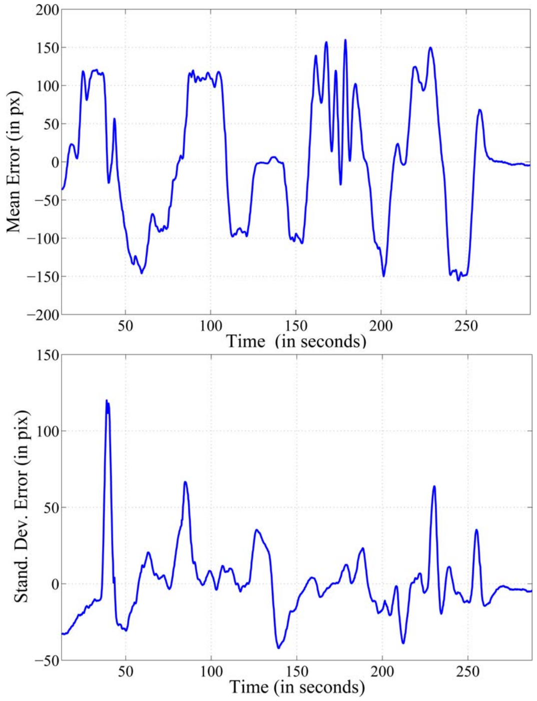

Fig. 1. Error in mean and variance for two moving targets.

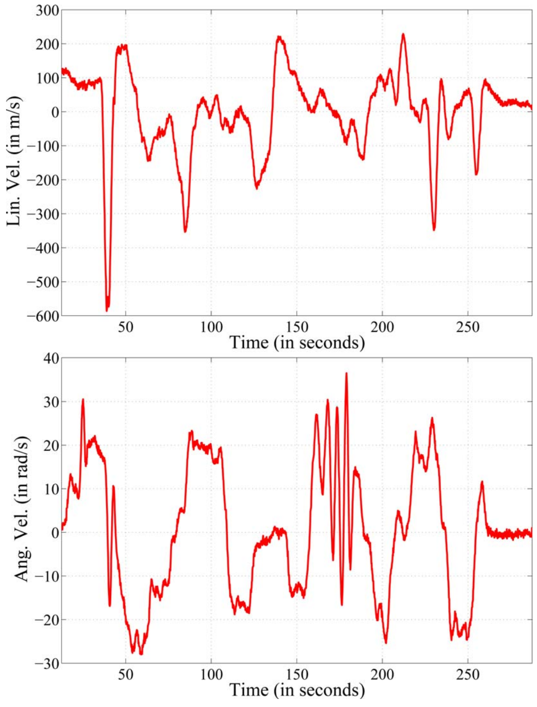

Fig. 2. Linear and Angular velocity for two moving targets.

VI. Conclusion and Future Work

This paper has introduced a novel method to track multiple objects and keep them in the camera FOV. A set of underdetermined task functions, which include the mean and variance of feature point coordinates, are used to solve for a feature point velocity that will keep features in the FOV. A third task function seeks to maximize motion perceptibility. There is no specific goal image or goal pose, rather the underdetermined nature of the task functions allows the camera to move as necessary to regulate the task function and keep objects in the FOV. This objective is in contrast with other visual servo controllers that require a specific goal. Furthermore, to the authors' knowledge, this is the first use of perceptibility in the feedback loop of a controller. Simulations of several object tracking tasks were performed to demonstrate this method.

There are several avenues of future work. There are numerous other task functions that could be used. For instance, it may be desirable to maintain a certain distance or orientation with respect to the tracked objects. The motion of the targets can be estimated using a variety of estimation or adaptive techniques, improving the performance of the system.

Acknowledgment

The authors would like to thank the reviwers and editors for their help in shaping this paper. They would also like to thank S. Hutchinson for his helpful insight and comments.

References

[1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

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

[3] N. R. Gans, G. Hu, and W. E. Dixon, "Keeping objects in the field of view: An underdetermined task function approach to visual servoing," in *Proc. IEEE Multi-Conf. Syst. Control*, 2008, pp. 432–437.

[4] N. Gans, G. Hu, and W. Dixon, "Keeping multiple objects in the field of view of a single PTZ camera," in *Proc. Amer. Control Conf.*, Jun. 2009, pp. 5259–5264.

[5] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control robots with visual feedback," *IEEE Trans. Robot. Automat.*, vol. RA-3, no. 5, pp. 404–417, Oct. 1987.

[6] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.

[7] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 3–15, 1987.

[8] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

[9] C. M. Bishop, A. Blake, and B. Marthi, "Super-resolution enhancement of video," presented at the Artif. Intell. Statist., Key West, FL, 2003.

[10] D. Stilwell, B. Bishop, and C. Sylvester, "Redundant manipulator techniques for partially decentralized path planning and control of a platoon of autonomous vehicles," *IEEE Trans. Syst. Man Cybern.*, vol. 35, no. 4, pp. 842–848, Aug. 2005.

[11] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1285–1292, Dec. 2006.

[12] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 985–994, Oct. 2009.

[13] R. Sharma and S. Hutchinson, "Motion perceptibility and its application to active vision-basedservo control," *IEEE Trans. Robot. Automat.*, vol. 13, no. 4, pp. 607–617, Aug. 1997.

[14] Y. Iwatani, "Task selection for control of active-vision systems," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 720–725, Aug. 2010.

[15] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. Robot. Automat.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.

[16] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 331–340, Apr. 2008.

[17] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 743–749, Jun. 2009.

[18] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 60–72, Feb. 2007.

[19] J. P. Barreto, L. Perdigoto, R. Caseiro, and H. Araujo, "Active stereo tracking of $nle3$ targets using line scan cameras," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 442–457, Jun. 2010.

[20] G. Hu, N. Gans, and W. E. Dixon, "Adaptive visual servo control," in in *Complexity and Nonlinearity in Autonomous Robotics, Encyclopedia of Complexity and System Science*. vol. 1, New York: Springer-Verlag, 2009, pp. 42–63.

[21] C. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-16, no. 1, pp. 93–101, Jan. 1986.

[22] H. Stark and J. Woods, *Probability and Random Processes with Applications to Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[23] B. Bishop, D. Stilwell, S. Eng, U. Acad, and M. Annapolis, "On the application of redundant manipulator techniques to the control of platoons of autonomous vehicles," in *Proc. IEEE Int. Conf. Control Appl.*, 2001, pp. 823–828.

[24] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, 1985.

[25] D. J. H. Garling, *Inequalities: A Journey into Linear Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[26] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal, "Continuous visual servoing despite the changes visibility in image features," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1214–1220, Dec. 2005.

Photometric Visual Servoing

Christophe Collewet and Eric Marchand

***Abstract*—This paper proposes a new way to achieve robotic tasks by two-dimensional (2-D) visual servoing. Indeed, instead of using classical geometric features such as points, straight lines, pose, or a homography, as is usually done, the luminance of all pixels in the image is considered here. The main advantage of this new approach is that it requires no tracking or matching process. The key point of our approach relies on the analytic computation of the interaction matrix. This computation is based either on a temporal luminance-constancy hypothesis or on a reflection model so that complex illumination changes can be considered. Experimental results on positioning and tracking tasks validate the proposed approach and show its robustness to approximated depths, low-textured objects, partial occlusions, and specular scenes. They also showed that luminance leads to lower positioning errors than a classical visual servoing based on 2-D geometric visual features.**

***Index Terms*—Cost function, optimization, photometry, visual features, visual servoing.**

Manuscript received July 15, 2010; revised November 24, 2010; accepted January 27, 2011. Date of publication March 17, 2011; date of current version August 10, 2011. This paper was recommended for publication by Associate Editor D. Kragic and Editor G. Oriolo upon evaluation of the reviewers' comments. This paper was presented in part at the 2008 and 2009 IEEE International Conference on Robotics and Automation and at the 2008 IEEE International Conference on Computer Vision and Pattern Recognition.

C. Collewet was with INRIA Rennes—Bretagne Atlantique, Lagadic Team, Rennes 35042, France. He is now with Cemagref, Rennes 35044, France (e-mail: christophe.collewet@cemagref.fr).

E. Marchand is with Université de Rennes 1, IRISA, INRIA Rennes—Bretagne Atlantique, Lagadic Team, Rennes 35042, France (e-mail: eric.marchand@irisa.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2011.2112593

I. Introduction

Visual servoing aims to control the motions of a robot by using data provided by a vision sensor [1]. More precisely, to achieve a visual-servoing task, a set of visual features has to be selected from the image to allow to control of the desired degrees of freedom (DOFs). A control law is then designed so that these visual features $\mathbf{s}$ reach desired values $\mathbf{s}^*$. The control principle is thus to regulate the error vector $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ to zero. To build the control law, the knowledge of the interaction matrix $\mathbf{L_s}$ is usually required [1].

Visual features are always designed from visual measurements $\mathbf{m}(\mathbf{p}_k)$ (where $\mathbf{p}_k$ is the camera pose at time k) that require a robust extraction, matching (between $\mathbf{m}(\mathbf{p}_0)$ and $\mathbf{m}(\mathbf{p}^*)$, where $\mathbf{p}^*$ is the desired camera pose), and real-time spatiotemporal tracking [between $\mathbf{m}(\mathbf{p}_{k-1})$ and $\mathbf{m}(\mathbf{p}_k)$]. However, this process is a complex task, as evinced by the abundant literature on the subject (see [2] for a recent survey), and is considered as one of the bottlenecks in the expansion of visual servoing. Thus, several works focus on alleviation of this problem. An interesting way to avoid any tracking process is to use nongeometric visual measurements as in [3] and [4] instead of geometric measurements, as is usually done. Of course, the direct use of nongeometric visual features also avoids any tracking process. In that case, parameters of a 2-D motion model have been used in [5]–[8]. Nevertheless, such approaches require a complex image processing task.

In this paper, we show that this tracking process can be totally removed and that no other information than the image intensity (the pure luminance signal) needs to be considered to control the robot motion. Indeed, to achieve this goal, we use as visual measurement and as visual feature the simplest that can be considered: the image intensity itself. We, therefore, call this new approach *photometric visual servoing*. In that case, the visual feature vector $\mathbf{s}$ is nothing but the image, while $\mathbf{s}^*$ is the desired image.

The image intensity as a feature has been considered previously [9], [10]. However, those works differ from our approach in two important points. First, they do not directly use the image intensity since an eigenspace decomposition is performed to reduce the dimensionality of image data. The control is then performed in the eigenspace and not directly based on the image intensity. Second, the interaction matrix related to the eigenspace is not computed analytically but learned during an off-line step. This learning process has two drawbacks: It has to be done for each new object and requires the acquisition of many images of the scene at various camera positions. We consider an analytical interaction matrix that avoids these issues. An interesting approach, which also directly considers the pixels intensity, has been recently proposed in [11]. However, only the translations and the rotation around the optical axis have been considered (that is the four most simple DOFs), whereas in our work, the six DOFs are controlled. However, an image processing step is still required. Our approach does not require this step.

In this paper, we summarize several previous works. In [12], the analytic computation of the interaction matrix related to the luminance for a Lambertian scene is provided, and only positioning tasks have been considered. In [13], this matrix has been computed considering a lighting source mounted on the camera and the use of the Blinn–Phong illumination model (a simplified version of the Phong model detailed in the next section), and only tracking tasks have been considered. In [14], the Phong model has been used, and only positioning tasks have been considered. In addition, these works refer to [15], where details concerning analytic computations are given. Note that in [16], although this is also a direct visual-servoing approach, the considered features used in the control law are very different. In this paper, we