2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009



# Keeping Multiple Objects in the Field of View of a Single PTZ Camera

N. R. Gans, G. Hu and W. E. Dixon

*Abstract*— **This paper introduces a novel visual servo controller designed to keep multiple moving objects in the camera field-of-view using a pan/tilt/zoom camera. In contrast to most visual servo controllers, there is no goal pose or goal image. In this paper, a set of task functions are developed to regulate the mean and variance of a set of image features. Regulating these task functions will deter feature points from leaving the camera field-of-view. An additional task function is used to maintain a high level of motion perceptibility, which ensures that desired feature point velocities can be achieved. To provide proper control of a pan/tilt/zoom camera, an image Jacobian is developed utilizing actuation of the focal length. Simulations of several object tracking tasks have verified the performance of the proposed method.**

## I. Introduction

The use of image information in feedback control, commonly referred to as visual servo (VS) control, is an established and diverse field. There are many approaches, including image based visual servoing (IBVS), position based visual servoing, partitioned methods and switching methods. See [1], [2] and the references therein for tutorials and historical discussion of these techniques. Most visual servo control methods have a common objective, to regulate a camera, robot manipulator, or unmanned vehicle to a desired pose or along a desired trajectory. However, many vision-based control tasks do not require a specific goal pose or trajectory.

One task that is not well characterized by a goal pose or image is keeping multiple, moving objects in the field-of-view (FOV). Consider the scenario of crowd surveillance. A camera views the crowd, and performs target segmentation and tracking methods to localize individuals of interest visible in the image. As the crowd moves and disperses, the method presented in this paper will move the camera in an attempt to keep all individuals in the FOV. A similar scenario involves tracking several unmanned vehicles and landmarks.

The method is rooted in classic IBVS [1]–[4], however there is no goal image or goal feature trajectory. Instead of task functions based on current and goal images, the proposed method regulates a series of low dimensional task functions based on current image features. The desired task function velocity is mapped to a time-varying feature velocity through corresponding task function Jacobians. These task function Jacobians are underdetermined and are suitable for task-priority kinematic control [3], [5], [6]. The time-varying image feature velocity is mapped to camera motions through IBVS methods. The resulting controller allows feature points to move within the image, and the camera will move to deter feature points from leaving the FOV.

Our initial developments in this field [7] introduced task functions based on mean, variance and perceptibility of feature point coordinates in the image. Please refer to [7] for a literature review, discussion and comparison with other visual servoing methods that use moments of feature points, as well as a discussion the use of task-function kinematic contro in mobile robot and visual servo control.

The work in [7] focused on the general case of a six degree-of-freedom (DOF), airborne mobile camera. The most prominent camera in surveillance tasks is the pan/tilt/zoom (PTZ) camera, which can rotate about two axes and alter its focal length to alter the perspective zoom of the image. In this current development, we extend our earlier efforts to include the unique kinematics of a PTZ camera, including the development of a 3DOF image Jacobian for feature points that includes the actuation of the focal length as a variable. A zoom term was appended to a standard image Jacobian in [8], though a closed-form expression for the zoom term is not given. A 3DOF image Jacobian corresponding to a PTZ camera watching a spherical target was developed in [9].

Sections II and III introduce background information related to the camera model and control approach. Sections IV-A through IV-C introduce the task functions used to generate the desired feature velocity. Simulations of tasks are given in Section V to demonstrate the performance of this controller.

## II. Camera Model and Image Jacobian for Pan/Tilt/Zoom Cameras

An image is a function of the relative pose between the scene and camera, and the camera intrinsic parameters. Consider a camera with coordinate frame $\mathcal{F}_c(t)$. The frame is oriented such that the camera optical axis corresponds to the z-axis of $\mathcal{F}_c(t)$, and the x-axis and y-axis are the horizontal and vertical directions of the image surface. The camera captures images of a collection of $k$ visible feature points. These points have Euclidean coordinates $M_i(t) \in \mathbb{R}^3$ defined as

$$M_i = [X_i, Y_i, Z_i]^T, \ \forall i \in \{1 \dots k\}$$

This research is supported in part by the NSF CAREER award CMS-0547448 and by the US Department of Energy grant number DE-FG04-86NE37967. This work was accomplished as part of the DOE University Research Program in Robotics (URPR). This research was performed, in part, while Nicholas Gans held a National Research Council Research Associateship Award at the Air Force Research Laboratory.

Nicholas Gans is with the National Research Council and Air Force Research Laboratory, Email: ngans@ufl.edu

Guoqiang Hu is with the Department of Mechanical and Nuclear Engineering, Kansas State University, Email: gqhu@ksu.edu

Warren Dixon Department of Mechanical and Aerospace Engineering, University of Florida, Email: wdixon@ufl.edu

978-1-4244-4524-0/09/$25.00 ©2009 AACC 

in the camera frame. The point $M_i(t)$ projects to a point in the feature point plane with coordinates

$$\frac{\lambda}{Z_i}M_i = \begin{bmatrix} \frac{\lambda X_i}{Z_i} \\ \frac{\lambda Y_i}{Z_i} \\ \lambda \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \lambda \end{bmatrix} = \begin{bmatrix} m_i \\ \lambda \end{bmatrix} \tag{1}$$

where $m_i(t) \in \mathbb{R}^2$ is the image coordinate of the $i^{\text{th}}$ feature point and $\lambda(t)$ is the focal length of the camera. Given the collection of $k$ feature points in the image, a feature state vector $m(t) \in \mathbb{R}^{2k}$ and state velocity vector $\dot{m}(t) \in \mathbb{R}^{2k}$ are defined as

$$\begin{aligned} m &= [x_1, y_1, \cdots, x_k, y_k]^T \\ \dot{m} &= [\dot{x}_1, \dot{y}_1, \cdots, \dot{x}_k, \dot{y}_k]^T. \end{aligned}$$

The extraction of such points could come from a variety of algorithms, such as tracking distinguishable points on moving targets [10], [11], the centroid of segmented blobs [12] or the corner points of bounding rectangle of tracked moving targets [13].

A PTZ camera cannot translate, but can rotate about two axes and alter the focal length of its lens. This work assumes that the center of rotation corresponds to the origin of the camera frame. Most visual servoing approaches assume a constant focal length, and many scale the system such that $\lambda = 1$. In the subsequent development here, a time-varying $\lambda(t)$ is introduced as an actuation to alter the image. Increasing $\lambda$ has the effect of zooming into the image, i.e. objects in the image will appear larger and feature points move toward the edge of the FOV. Decreasing $\lambda$ will zoom out of the image, i.e. objects will appear smaller and feature points will move toward the image center. While this is similar to the effect of translating along the camera optical axis, there are important differences. Most notably, translation is not commutative with rotation, but zoom is commutative. Incorporating $\lambda$ into a VS control system requires developing a novel image Jacobian.

If the viewed objects are not moving in the world frame, the derivative of $M_i(t)$ as a function of camera velocity is given by

$$\dot{M}_i = -\omega \times M_i \tag{2}$$

where $\omega(t) = [\omega_x,\ \omega_y,\ 0]^T \in \mathbb{R}^3$ is the angular velocity of the camera. Equation (2) can be rewritten as

$$\begin{bmatrix} \dot{X}_i \\ \dot{Y}_i \\ \dot{Z}_i \end{bmatrix} = \begin{bmatrix} -Z_i\omega_y \\ Z_i\omega_x \\ X_i\omega_y - Y_i\omega_x \end{bmatrix}. \tag{3}$$

The derivative of (1) is given by

$$\frac{d}{dt}\left(\frac{\lambda}{Z_i}M_i\right) = \frac{\lambda}{Z_i^2}\left(\dot{M}_i Z_i - M_i\dot{Z}_i\right) + \frac{\dot{\lambda}}{Z_i}M_i. \tag{4}$$

Combining the terms in (3) with the top two rows (4) gives

$$\begin{aligned} \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} &= \frac{1}{\lambda}\begin{bmatrix} x_i y_i \omega_x - (\lambda^2 + x_i^2)\omega_y + \dot{\lambda}x_i \\ (\lambda^2 + y_i^2)\omega_x - x_i y_i \omega_y + \dot{\lambda} y_i \end{bmatrix} \\ &= \frac{1}{\lambda}\begin{bmatrix} x_i y_i & -(\lambda^2 + x_i^2) & x_i \\ (\lambda^2 + y_i^2) & -x_i y_i & y_i \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \dot{\lambda} \end{bmatrix} \\ &= L_{i\lambda} v_{ptz} \end{aligned} \tag{5}$$

where $L_{i\lambda}(t)$ is the image Jacobian for point $i$ for a PTZ camera, and $v_{ptz}(t) \in \mathbb{R}^3 = \left[\omega_x,\ \omega_y,\ \dot{\lambda}\right]^T$. It can be seen that $L_{i\lambda}(t)$ matches the first three rows of the Jacobian for a sphericle target in [9] The time derivative of a feature point vector $m(t)$ is given by

$$\dot{m} = L_\lambda v_{ptz}$$

where $L_\lambda(t) \in \mathbb{R}^{2k\times 3} = \left[L_{1\lambda}^T, \cdots, L_{k\lambda}^T\right]^T$. In the case that the feature points are not static in an inertial world frame (e.g., the feature points are tracked on moving objects), the time derivative of the feature points is given by

$$\dot{m} = L_\lambda v_{ptz} + \varepsilon \tag{6}$$

where $\varepsilon(t)$ is an unknown function caused by the targets motion. We make a practical assumption that $\varepsilon(t)$ is bounded.

The first two columns of $L_\lambda(t)$ are the same as the fourth and fifth columns of the classic six -DOF image Jacobian [1], [3]. One notable feature of $L_\lambda(t)$ is that it is not dependent on feature depths $Z_i(t)$. The six-DOF image Jacobian includes the depths $Z_i(t)$ as variables. The feature point depths must be known or accurately estimated for six-DOF IBVS, and IBVS is brittle to depth estimation errors [14]. As seen in (5), the depths do not need to be known, which increases robustness and decreases computational complexity.

## III. Task Function-Based Kinematic Control

The control objective is to keep a set of feature points within the camera FOV without a specific camera pose. Motivated by this objective, a set of task functions are defined using image feature coordinates. By regulating the task functions, feature points can be inhibited from leaving the FOV. However, the task functions may compete in the sense that reducing the error of one task function may increase the error of another. To avoid competition between task functions, task-priority kinematic control [5], [6] is used.

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

$$\phi = f(m)$$

with derivative

$$\dot{\phi} = \sum_{i=1}^{k}\frac{\partial f}{\partial m_i}\dot{m}_i = J(m)\dot{m} \tag{7}$$

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

The task is to drive the feature points along a desired velocity $\dot{m}_\phi(t)$ 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 based on the minimum-norm approach [6] is given as

$$\begin{aligned} \dot{m}_{\phi} &= J^{\dagger}\left[\dot{\phi}_d - \gamma(\phi - \phi_d)\right] \\ &= J^T\left(JJ^T\right)^{-1}\left[\dot{\phi}_d - \gamma(\phi - \phi_d)\right] \end{aligned} \tag{8}$$

where $\gamma$ is a positive scalar gain constant, and $J^{\dagger}(m) \in \mathbb{R}^{2k\times l}$ denotes the minimum-norm general inverse of $J(m)$. Based on (8), the camera velocity $v_{ptz}(t)$ is designed as

$$v_{ptz} = L_{\lambda}^{+}\dot{m}_{\phi} = \left(L_{\lambda}^T L_{\lambda}\right)^{-1} L_{\lambda}^T \dot{m}_{\phi} \tag{9}$$

where $L_{\lambda}^{+}(t) \in \mathbb{R}^{3\times 2k}$ denotes the least squares general inverse of $L_{\lambda}(t)$. Note that $J(m)$ is underdetermined (i.e. more columns than rows), and $L_{\lambda}(t)$ is overdetermined (i.e. more rows than columns). Therefor, the general inverses have different solutions and are denoted differently.

A set of task functions can be combined in various ways. The simplest approach is to add the desired feature velocity from each task function to create the total desired feature point velocity. For example, consider two tasks $\phi_a(t)$ and $\phi_b(t)$ with resulting desired feature velocities $\dot{m}_a(t)$ and $\dot{m}_b(t)$. The camera velocity is then given as

$$v_{ptz} = L_{\lambda}^{+}\left(\dot{m}_a + \dot{m}_b\right).$$

Since the task function velocities are undetermined, an optional method is to choose one task as primary, and project the other tasks into the nullspace of the primary task derivative [5], [6] as

$$\begin{aligned} v_{ptz} &= L_{\lambda}^{+}\left(\dot{m}_a + (I - J_a^{\dagger}J_a)\dot{m}_b\right) \\ &= L_{\lambda}^{+}\left(J_a^{\dagger}\dot{\phi}_a + (I - J_a^{\dagger}J_a)J_b^{\dagger}\dot{\phi}_b\right) \end{aligned} \tag{10}$$

where $J_a(m_a)$ and $J_b(m_b)$ are the task Jacobian matrices with respect to $\phi_a(t)$ and $\phi_b(t)$, respectively.

The approach in (10) 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 presented as part of a task-priority kinematic controller. Two task functions are designed to regulate the mean and variance of the feature point coordinates. Regulating the mean at the camera center will keep the feature points centered in the FOV. Regulating the variance will restrict the distance between the feature points and keep feature points away from the edge of the FOV. A third task function maximizes motion perceptibility, which ensures desired image velocities can be met. These task functions are cascaded through nullspace projection and mapped to camera velocity, as described Section III. Cascading controllers for mean and perceptibility with a controller for variance will deter objects from leaving the FOV.

A benefit of the mean and variance task functions is that no feature point tracking, matching or registration needs to occur, and the order of feature points in the feature vector $m(t)$ is not important. This benefit is due to the fact that all specific point information is lost by taking the mean and variance. Some feature extraction methods, such as corner detection or taking the centroid of blob segments, do not match or order features between images without additional algorithms. Matching can be a difficult problem, so this is a notable advantage.

If variance is regulated to a suitably small goal value, a subset of feature points can be guaranteed to stay in view. Given a distribution with mean $\bar{x}$ and variance $\sigma_x^2$, Chebyshev's inequality states [15] that no more than $1/k^2$ of the values are more than $k$ standard deviations away from the mean. Specifically, Chebyshev's inequality states 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. For a normally distributed random process, these limits are tighter such that approximately 95% of all values will be within two standard deviations, and 99.7% of all values will be within three standard deviations. Consider a camera with a 512x512 pixel FOV regulating the variance to $86^2$ will ensure that at least 89% of all points are in the FOV. Features may be lost due to leaving the FOV, occlusions, or failure of the image processing routine that extracts features. If this occurs, the last known coordinates can be used to attempt to recapture it at a later time, or the point can be discounted from the mean and regulation continues with the remaining points.

### A. Task Function for the Mean of the Image Points

Controlling the mean of the feature point coordinates helps to ensure the feature points are centered around a position in the image plane. Let $\phi_m(t) \in \mathbb{R}^2$ denote a task function 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

$$\begin{aligned} \dot{\phi}_m &= \frac{1}{k}\sum_{i=1}^{k}\frac{\partial \phi_m}{\partial m_i}\dot{m}_i = J_m\dot{m} \\ &= J_m\left(L_{\lambda}v_{ptz} + \varepsilon\right) \end{aligned}$$

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

$$J_m = \frac{1}{k}\left[I_2, \cdots, I_2\right] \tag{11}$$

where $I_2$ is the $2\times 2$ identity matrix and is repeated $k$ times.

In the general case, the objective is to regulate the mean to track a desired trajectory, helping to ensure the feature points are centered around a specific, time-varying point in the image. Define a desired task function trajectory $\phi_{md}(t)$, with a known, smooth derivative $\dot{\phi}_{md}(t)$. PID control can be used to generate a feature velocity that will track the desired value of $\phi_{md}(t)$. This feature velocity is denoted $\dot{m}_m(t) \in \mathbb{R}^{2k}$, and is given by

$$\begin{aligned}\dot{m}_m &= -J_m^{\dagger}\left(\gamma_{mp}\phi_{me} + \gamma_{mi}\int_0^t \phi_{me}dt\right. \\ &\qquad \left. +\gamma_{md}\frac{d}{dt}\phi_{me} - \dot{\phi}_{md}\right) \qquad (12)\end{aligned}$$

where $\phi_{me}(t) = \phi_m(t) - \phi_{md}(t)$ and $\gamma_{mp}, \gamma_{mi}, \gamma_{md} \in \mathbb{R}$ are constant gains. If the desired trajectory $\phi_{md}(t)$ is simplified to a constant set value $\phi_{md}$, equation (12) can be simplified as

$$\dot{m}_m = -J_m^{\dagger}\left(\gamma_{mp}\phi_{me} + \gamma_{mi}\int_0^t \phi_{me}dt + \gamma_{md}\frac{d}{dt}\phi_{me}\right).$$

### *B. Task Function for the Variance of the Image Points*

Regulating the variance of the feature point coordinates will regulate the spread of the feature points in the image. That is, regulating the variance will control how far the feature points drift from the mean value. A task function $\phi_v(t) \in \mathbb{R}^2$ is given by the sample variance

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

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

To find the variance task function Jacobian, consider the partial derivative of $\phi_v(t)$ with respect to $x_1(t)$

$$\begin{aligned}\frac{\partial \phi_v}{\partial x_1} &= \frac{\partial}{\partial x_1}\left[\frac{1}{k}\sum_{i=1}^{k}(x_i - \bar{x})^2\right] \\ &= \frac{2}{k}\left[(x_1 - \bar{x}) - \frac{1}{k}\sum_{i=1}^{k}(x_i - \bar{x})\right] \\ &= \frac{2}{k}\left[x_1 - \bar{x}\right].\end{aligned}$$

Repeating the above simplifications for all $x_i(t),\ y_i(t),\ i \in \{1 \ldots k\}$, gives the time derivative of $\phi_v(t)$ as

$$\begin{aligned}\dot{\phi}_v &= J_v\dot{m} \\ &= J_v\left(L_\lambda v_{ptz} + \varepsilon\right)\end{aligned}$$

where $L_\lambda(t)$, $v_{ptz}(t)$ and $\varepsilon(t)$ were given in (6), and $J_v(t) \in \mathbb{R}^{2\times 2k}$ is a task function Jacobian given by

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

To regulate the variance to a desired trajectory $\phi_{vd}(t)$ (with a known, smooth derivative $\dot{\phi}_{vd}(t)$) the feature point velocity $\dot{m}_v(t) \in \mathbb{R}^{2k}$ can be designed by following the method in Section IV-A to give

$$\dot{m}_v = -J_v^{\dagger}\left(\gamma_{vp}\phi_{ve} + \gamma_{vi}\int_0^t \phi_{ve}dt + \gamma_{vd}\frac{d}{dt}\phi_{ve} - \dot{\phi}_{vd}\right) \qquad (13)$$

where $\phi_{ve}(t) = \phi_v(t) - \phi_{vd}(t)$ and $\gamma_{vp}, \gamma_{vi}, \gamma_{vd} \in \mathbb{R}$ are constant gains.

An important consequence of using mean and variance as task functions is that the control laws for mean and variance do not interfere with each other**.** This means 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 equation (10).

**Theorem 1:** Given the Jacobian matrices $J_m$ and $J_v$, $J_mJ_v^{\dagger} = J_vJ_m^{\dagger} = 0$.

**Proof:** The pseudo inverse $J_v^{\dagger}(t) \in \mathbb{R}^{2k\times 2}$ can be written in closed form and multiplied with (11) to give

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

It can similarly be shown that $J_vJ_m^{\dagger} = 0$. $\square$

Combining (10), (12) and (13) and the result of Theorem 1, the resulting feature point velocity from prioritizing the tasks is given by

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

### *C. Task Function for Perceptibility of Image Points*

Sharma and Hutchinson presented the concept of motion perceptibility in [16]. 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 due to 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 $L_\lambda(t)$, defined as

$$w_v = \sqrt{\det(L_\lambda^T L_\lambda)} = \prod_{i=1}^{3}\sigma_i$$

where $\sigma_i(t) \in \mathbb{R}^+$ are the singular values of $L_\lambda(t)$. Maximizing $w_v(t)$ is accomplished by maximizing each $\sigma_i(t)$. The matrix $L_\lambda^T(t)L_\lambda(t) \in \mathbb{R}^{3\times 3}$ is positive definite and symmetric, so the eigenvalues of $L_\lambda^T(t)L_\lambda(t)$ are equal to $\sigma_i^2(t)$. The trace of a matrix is equal to the sum of its eigenvalues. Therefore, the trace of $L_\lambda^T(t)L_\lambda(t)$ is related to the singular values by

$$Tr(L_\lambda^T L_\lambda) = \sum_{i=1}^{3} \sigma_i^2.$$

Increasing the trace of $L_\lambda^T(t)L_\lambda(t)$ will also increase the perceptibility.

The trace of $L_\lambda^T(t)L_\lambda(t)$ is given by

$$\begin{aligned} Tr(L_\lambda^T L_\lambda) &= \sum_{i=1}^{k} \Big( 2x_i^2 y_i^2 + \left(y_i^2 + \lambda\right)^2 \\ &\quad + \left(x_i^2 + \lambda\right)^2 + x_i^2 + y_i^2 \Big) \end{aligned}$$

A task function for perceptibility can be defined as a function of feature-point coordinates as

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

Since it is desired to increase $Tr(L_\lambda^T L_\lambda)$, regulating $\phi_p(t)$ to 0 will result in increasing the trace. The time derivative of $\phi_p(t)$ is given by

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

where $L_\lambda(t)$, $v_{ptz}(t)$ and $\varepsilon(t)$ were given in (6), and $J_p(m) \in \mathbb{R}^{1\times 2k}$ is the task function Jacobian for perceptibility. The matrix $J_p(m)$ is undefined only for the nongeneral case that $\forall i,\ m_i = 0$.

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

$$\dot{m}_p = -\gamma_p J_p^\dagger \phi_p \tag{14}$$

where $\gamma_p$ is a positive scalar gain constant. As in the mean and variance control laws, a tracking error will exist for the perceptibility task. However, the use of integral feedback is not recommended for the perceptibility regulation due to the fact that $\phi_p(t)$ is unlikely to ever become zero, leading to possible integrator windup and related stability problems.

### D. Cascaded Camera Control Law

The control objective is to design a kinematic controller for a PTZ camera that maintains a set of feature points within the camera FOV. The mean of feature point coordinates is the best variable to measure the center of the feature points. As shown in Theorem 1, regulating the mean and variance will not conflict, so they are chosen as the primary tasks in order to keep the feature points centered in the FOV and to restrict the distance between the feature points and the image center. These two tasks will deter feature points from leaving the FOV. High perceptibility will allow these tasks to work more efficiently by ensuring 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 feature velocities given in (12), (13), and (14) are used in the nullspace projection camera velocity (10) to give the overall controller as

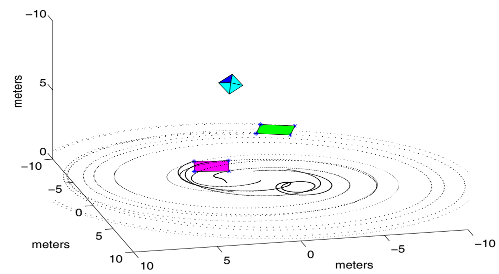


Fig. 1. 3D object motion and camera.

$$\begin{aligned} v_{ptz} &= L_\lambda^+ \left(\dot{m}_m + \dot{m}_v + \left(I - J_m^\dagger J_m\right)\left(I - J_v^\dagger J_v\right)\dot{m}_p\right) \\ &= L_\lambda^+ \left(\dot{m}_m + \dot{m}_v + \left(I - J_m^\dagger J_m - J_v^\dagger J_v\right)\dot{m}_p\right) \quad (15) \end{aligned}$$

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

## V. Simulation Results

Simulations are performed using the PID controller given in (12)-(15). Simulations have been run in a realistic virtual environment as well, requiring feature extraction and real time image and control processing. Movies of these simulations are available at [17]. In this simulation, a camera observes two rigid, square objects with dimensions of 1m×1m. The centers of the two objects are initially located at coordinates $[X, Y, Z]^T = [1, 1, 0]$ and $[X, Y, Z]^T = [-1, -1, 0]$ in the world frame. The objects move independently and the corner of the two squares give eight feature points to track. This simulation mimics the case of a camera tracking corners on two vehicles. The camera is fixed at coordinates $[11, 4.5, 12]^T$ in the world frame. The mean was regulated to the image center. The variance was regulated to $[100^2,\ 100^2]^T$, i.e. a standard deviation of 100 pixels. The gains were selected as $\gamma_{mp} = 0.05$, $\gamma_{mi} = 7 \times 10^{-3}$, $\gamma_{md} = 0.07$, $\gamma_{vp} = 7.5 \times 10^{-5}$, $\gamma_{vi} = 3.5 \times 10^{-7}$, $\gamma_{vd} = 3.5 \times 10^{-6}$, $\gamma_p = 5 \times 10^{-5}$. The simulation was executed for 16 seconds at 30 frames per second. The square objects moved with sinusoidal velocities along all six degrees of freedom.

Fig. 1 shows a third person view of the two objects and camera. The 3D paths of the corners points are shown as dotted lines. Fig. 2 depicts the trajectory of the two objects and feature points in the image plane. The feature points all remain in the FOV. The dashed ellipse and square represent the final values of the variance and mean, while the solid ellipse and star represent the goal variance and mean. Fig. 3 shows the task function error over time. The mean error is well regulated about zero. While variance is periodic, it remains bounded about zero error. Fig. 4 shows the elements of the camera velocity.

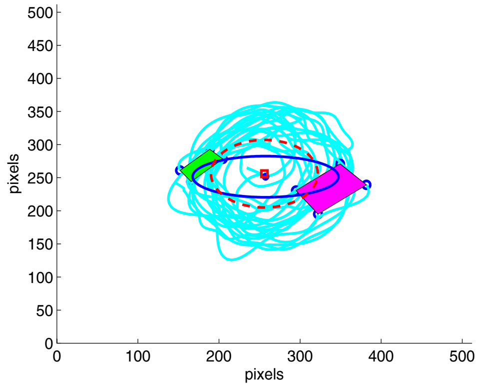


Fig. 2. Object trajectories in the image when regulating variance.

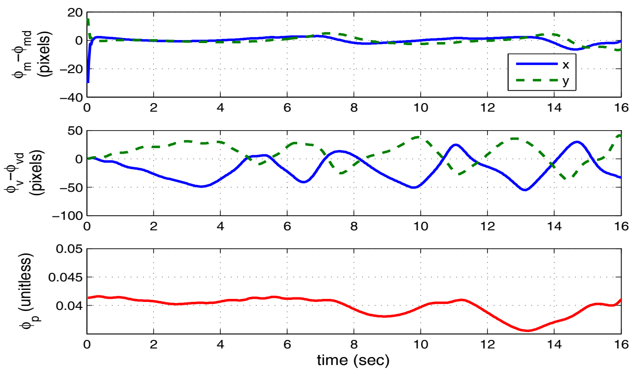


Fig. 3. Task function errors while tracking two moving targets.

## VI. Conclusions and Future Work

This paper introduces a method to track multiple moving objects and keep them in the camera FOV using a Pan/Tilt/Zoom camera. A set of underdetermined task functions of sample mean and variance of feature point coordinates are used to deter feature points from leaving the FOV. A third task function seeks to maximize motion perceptibility. There is no specific goal image or goal pose. The underdetermined nature of the task functions allows the camera to move as necessary to regulate the task functions and inhibit feature points from approaching the edge of the FOV. This objective is in contrast to other visual servo controllers that require a specific goal image or goal pose. 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. To accommodate the use of a PTZ camera, a novel image Jacobian for feature points was developed that includes focal length as an actuation. One notable benefit of the PTZ camera is that it does not depend on the depth of feature points, like traditional image Jacobians.

There are several avenues of future work. Simulations are very promising, but a proper stability analysis will be performed. The motion of the targets can be estimated using a variety of estimation or adaptive techniques, improving the performance of the system. Experimental analysis is also underway, which will include target recognition, localization and interframe tracking.

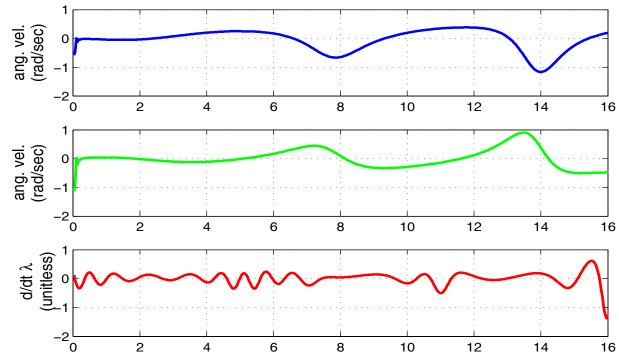


Fig. 4. Camera velocity while tracking two moving targets.

## References


[1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 651–670, Oct. 1996.

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

[3] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 313–326, June 1992.

[4] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control robots with visual feedback," *IEEE Trans. Robot. Automat.*, vol. 3, pp. 404–417, Oct. 1987.

[5] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robotics Research*, vol. 9, pp. 3–15, 1987.

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

[7] 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. Systems and Control*, pp. 432–437, 2008.

[8] K. Hosoda, H. Moriyama, and M. Asada, "Visual servoing utilizing zoom mechanism," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 178–183, 1995.

[9] K. Hatano and K. Hashimoto, "Image-based visual servo using zoom mechanism," in *Proc. SICE Annual Conference*, vol. 3, pp. 2443–2446 Vol.3, Aug. 2003.

[10] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[11] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Computer Vision*, vol. 2, pp. 1150–1157, 1999.

[12] M. Niethammer, A. Tannenbaum, and S. Angenent, "Dynamic active contours for visual tracking," *IEEE Trans. Automat. Contr.*, vol. 51, pp. 562–579, April 2006.

[13] C. Gentile, O. Camps, and M. Sznaier, "Segmentation for tracking in the presence of severe occlusion," *IEEE Trans. Image Processing*, vol. 2, pp. 166–178, Feb. 2004.

[14] E. Malis and P. Rives, "Robustness of image-based visual servoing with respect to depth distribution errors," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, pp. 1056–1061, 2003.

[15] H. Stark and J. Woods, *Probability and Random Processes with Applications to Signal Processing*. Prentice Hall, 2002.

[16] 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, 1997.

[17] http://s5.photobucket.com/albums/y176/ngans/VS Videos/PTZ VS/.