Visual Servoing to an Arbitrary Pose with Respect to an Object Given a Single Known Length[1]

N. R. Gans, A. P. Dani and W. E. Dixon

Department of Mechanical Aerospace Engineering

University of Florida, Gainesville, FL USA

ngans@ufl.edu, ashwin31@ufl.edu, wdixon@ufl.edu

Abstract— A method is presented to attach reference frames to piecewise planar objects in view a single camera. This method uses Euclidean homography relationships and a single known geometric length on a single object. By attaching reference frames to objects in the scene, the method is useful in position-based visual servo control, where it allows control of pose with respect to an object. The method is distinguished from methods that require a detailed model of the object/scene to give camera pose relative to an object, and it is distinguished from methods that can only give current camera pose with respect to a pose where a reference image was taken. Simulations of the method for camera-in-hand and camera-to-hand visual servo control tasks are presented. Experiments are presented where the reconstruction method is used to estimate the pose of a vehicle. These experiments represent the initial steps in a vision-based vehicle following controller.

I. INTRODUCTION

Position-based visual servo (PBVS) control methods [1]–[4] use three dimensional scene information that is reconstructed from image information. That is, the camera acts as a "Cartesian sensor", where pose estimation algorithms use camera data to generate an error signal in Cartesian space. This error signal is then used in a feedback control law. Reconstruction methods vary in terms of computational complexity, accuracy, and the amount of information required. Methods exist to determine the camera's pose (i.e. the camera's position and orientation) with respect to an object [5], [6]; however, these methods require accurate geometric knowledge of the object, i.e. knowledge of the relative coordinates of all 3D points, lines, etc. used in the algorithm. These methods might be called "hard PBVS" methods.

An accurate model is not available for unknown scenes and may be hard to obtain, even for a known object. Alternately, given two images it is possible to use the Essential Matrix or Homography matrix [7], [8] to determine the rotation and translation between the associated camera poses. Essential or Homography Matrices have been used in numerous control designs [9]–[11]. Typically, these methods could only move to a goal pose where an image had been captured, and are often referred to as *teach by showing* methods. Since

Image-Based Visual Servoing (IBVS) [2], [12] methods are typically teach by showing, these PBVS methods have some commonality with IBVS and could be called "soft PBVS" methods. The major drawback of these methods is that they can only regulate the pose of the camera with respect to a reference pose where some reference image was taken. They are not suitable to regulate the pose of the camera with respect to a viewed object or the pose of one viewed object with respect to another.

This paper presents a new method to attach a reference frame to a planar surface containing feature points. This method uses the Euclidean Homography derived from two images and requires knowledge of a single geometric length between two visible points on the plane. Alternately, if a length on the object is unknown, but the robot is well calibrated (i.e. accurate measurements of the robot pose and velocity are available), it is possible to attach the reference frame using a known camera motion rather than a known length.

The attached reference frame allows feedback control to move the camera to an arbitrary position relative to the object. This has numerous applications such as manipulation and docking. Furthermore, without any further geometric knowledge, it is possible to attach a reference frame to every other static planar surface. There is no requirement that the scene remain static after attaching the reference frames, so it becomes possible to track or control moving objects after a brief learning phase. The presented method has advantages over both the "hard" and "soft" PBVS methods. Knowledge of a single geometric length is much less restrictive than the complete geometric knowledge of all lengths needed for "hard" pose recovery methods such as [5], [6], and the ability to achieve a pose with respect to an object rather than a reference pose is more useful for some tasks than "soft" methods.

Silveira, et al. [13] recently explored the ability to visual servo a camera to an arbitrary pose. This method also makes use of the Euclidean Homography and is capable of positioning the camera outside its initial field of view. However, there are major differences between the approach in [13] and the approach presented in this paper. The method by Silveira, et al. defines the arbitrary pose relative to a reference pose where a reference image was taken. The method presented here allows the camera to move to an arbitrary pose relative to a visible object, rather than a

reference pose. In order to achieve an arbitrary pose with respect to an object, the method by Silveira, et al. requires a priori knowledge of the relative transformation between the reference pose and the object. Furthermore, the method presented by Silveira, et al. decomposes the Homography to recover rotation and translation information, while the method presented in this paper uses the Homography matrix only to retrieve the normal to the plane. The normal vector is used in conjunction with a known geometric length to attach a reference frame to the plane.

The method presented in the paper is similar in spirit to [14], where an image-based method was presented to align a camera perpendicular to a planar surface. Given a calibrated robot, odometry information can generate an estimate of object depth and achieve a desired pose with respect to the object.

In addition, the method presented in this paper can provide priori information required by some vision-based estimation algorithms. For instance, [15] presents a homography-based velocity estimator, which requires knowledge of a geometric length and knowledge of the pose of the object in the initial image frame. Using the reconstruction method we introduce here, all the necessary information is recovered from just the single geometric length. A variation of this method is used in [16] to estimate the position of feature points if they become occluded or temporarily left the field of view.

The paper is organized as follows. Section II, presents background information introduces notation. The reconstruction method for the baseline case of a single set of coplanar feature points is presented in Section III. Section IV extends the developed method to the case of piecewise planar objects or piecewise planar scenes. Experimental and simulated results are presented in Section V. Finally, Section VI provides a discussion of several applications and extensions of this technique that we are actively pursuing.

II. Background

Consider a camera with reference frame $\mathcal{F}_c^*$. The camera views a collection of $k \geq 4$ feature points lying in a plane π_s in front of the camera. These points have in the ordinates $\bar{m}$, $\bar{m}_j^* \in \mathbb{R}^3$ in the camera reference frame given as

$$\bar{m}_j^* = [x_j^*, y_j^*, z_j^*]^T, \ \forall j \in \{1 \dots k\}.$$

An image of the points is captured, resulting in a projection to a set of points in the image plane π_i. These image points are given by the normalized coordinates $m_j^* \in \mathbb{R}^3$ given as

$$m_j^* = [\frac{x_j^*}{z_j^*}, \frac{y_j^*}{z_j^*}, 1]^T, \ \forall j \in \{1 \dots k\}.$$

The plane π_s has normal vector $-n^* \in \mathbb{R}^3$ in $\mathcal{F}_c^*$, while π_i has a normal coincident with the $-z$-axis of $\mathcal{F}_c^*$. The constant, scalar distance

$$\bar{s}_1 = \|\bar{m}_1^* - \bar{m}_2^*\| \tag{1}$$

is assumed to be known. These coordinate frame relationships are illustrated in Fig. 1.

By moving the camera (similarly, moving π_s) by a translation $x(t) \in \mathbb{R}^3$ and rotation $R(t) \in SO(3)$, the camera will obtain a new pose $\mathcal{F}_c(t)$. The points have Euclidean and normalized coordinates $\bar{m}_j(t)$, $m_j(t) \in \mathbb{R}^3$ in $\mathcal{F}_c(t)$ given by

$$\bar{m}_j(t) = [x_j(t), y_j(t), z_j(t)]^T, \ \forall j \in \{1 \dots k\}$$
$$m_j(t) = [\frac{x_j(t)}{z_j(t)}, \frac{y_j(t)}{z_j(t)}, 1]^T, \ \forall j \in \{1 \dots k\}.$$

A homography exists that maps m_j^* to $m_j(t)$. This homography can be defined as $H(t) \in \mathbb{R}^{3 \times 3}$ such that

$$m_j = \frac{z_j^*}{z_j} H m_j^* \tag{2}$$

$$m_j = \alpha \left(R + \frac{x}{d^*} n^{*T} \right) m_j^*, \tag{3}$$

where $\alpha_j = \frac{z_j^*}{z_j}$ is a scalar depth ratio. The matrix $H(t)$ is decomposed to recover $R(t)$, $\frac{x(t)}{d^*}$, n^* and α_j, $\forall j \in \{1 \dots k\}$ [7], [17]. Note that translation is only recovered up to a scaled factor $x_d(t) = \frac{x(t)}{d^*}$, and the depth d^* is generally unknown.

Using standard projective geometry, the Euclidean coordinate $\bar{m}_j(t)$ is expressed in image-space pixel coordinates as $p_j(t) = [u_j(t), v_j(t), 1]^T$. The relationship between $p_j(t)$ and $m_j(t)$ is given by

$$p_j = A m_j, \tag{4}$$

where $A \in \mathbb{R}^{3 \times 3}$ is a constant, invertible, upper-triangular camera calibration matrix [8]. Using (4), the Euclidean relationship in (3) is expressed as

$$p_j = \alpha_j A H A^{-1} p_j^*$$
$$= \alpha_j G p_j^*. \tag{5}$$

Given knowledge of A and $k \geq 4$, it is possible to solve a set linear equations for $G(t)$ and recover $H(t)$, $R(t)$, $\frac{x(t)}{d^*}$, n^* and $\alpha_j(t)$.

III. Geometric Reconstruction of Coplanar Feature Points

A. Attaching a Reference Frame to a Planar Object

The Euclidean homography algorithm is sufficient to estimate the rotation and scaled translation of the camera or object with respect to some reference pose. However, it is not sufficient to solve for the pose of the camera with respect to a viewed object or the pose of one viewed object with respect to another. The following development provides a method where the Euclidian homography algorithm is used along with some additional information to attach a reference frame $\mathcal{F}_s$ to a planar object, with rotation and translation given with respect to the camera frame $\mathcal{F}_c$. That is, the development in this section is used to determine the rotation and translation between $\mathcal{F}_s$ and $\mathcal{F}_c$.

Consider a camera viewing a planar object with four or more distinguishable feature points, and denote the feature point plane as π_s. If the camera and/or object move over

Fig. 1.   Elements of the reconstruction method

time, a set of linear equations can be solved for the Homography matrix $H(t)$ as in (2). The matrix $H(t)$ is decomposed into $R(t), \frac{x(t)}{d^*}$, and n^* as in (2), where the reference frame $\mathcal{F}_c^*$ can be taken as $\mathcal{F}_c(t_0)$, i.e., the initial frame. The normal vector of the planar object in the current frame $\mathcal{F}_c$ is given as

$$n(t) = R(t)n^*.$$

The goal is to attach a reference frame $\mathcal{F}_s(t)$ to the planar object. Without loss of generality, the origin of $\mathcal{F}_s(t)$ is assigned to the point $\bar{m}_1$. The orthonormal vectors i_x, i_y, $i_z \in \mathbb{R}^3$ that define $\mathcal{F}_s(t)$ form a rotation matrix $R(t)$ in $\mathcal{F}_c$ as

$$R = \begin{bmatrix} i_x & i_y & i_z \end{bmatrix}. \tag{6}$$

The columns of $R(t)$ in (6) are defined as

$$i_z = -n \tag{7}$$

$$i_x = \frac{\bar{m}_2 - \bar{m}_1}{\bar{s}_1} \tag{8}$$

$$i_y = -n \times \frac{\bar{m}_2 - \bar{m}_1}{\bar{s}_1}, \tag{9}$$

where the constant distance between the two feature points $\bar{s}_1 = \|\bar{m}_1 - \bar{m}_2\|$ is assumed to be known. If $\bar{m}_1(t)$ and $\bar{m}_2(t)$ were known, then i_x and i_y can be determined from (8) and (9) since $\bar{s}_1$ is assumed to be known. To solve for $\bar{m}_1(t)$ and $\bar{m}_2(t)$, a new plane π_s' is defined with normal $-n(t)$ (so π_s' is parallel to π_s) and containing the normalized image point $m_1(t)$. A line l is defined from the origin of $\mathcal{F}_c$ through $m_2(t)$ and $\bar{m}_2(t)$. The plane π_s' intersects l at a point m_2'. The unknown distance between $m_1(t)$ and $m_2'(t)$ is s_1, as illustrated in Fig. 1.

The primitives l and π_s' are defined by the sets of points $q \in \mathbb{R}^3$ that satisfy the implicit functions

$$l = \{q \mid q - um_2 = 0, \forall u \in \mathbb{R}\} \tag{10}$$

$$\pi_s' = \{q \mid n \cdot (q - m_1) = 0, \ q, n, m_1 \in \mathbb{R}^3\}. \tag{11}$$

The intersection of π_s' and l occurs when

$$u = \frac{n \cdot m_1}{n \cdot m_2}. \tag{12}$$

The expressions in (10) and (12) are combined to solve for the point $q = m_2'$ as

$$m_2' = \frac{n \cdot m_1}{n \cdot m_2} m_2.$$

The solution for $m_2'(t)$ is used to solve for s_1 as

$$s_1 = \|m_2' - m_1\| \tag{13}$$

and the properties of similar triangles is used to calculate the following:

$$\frac{s_1}{\bar{s}_1} = \frac{\|m_1\|}{\|\bar{m}_1\|} = \frac{\left\|m_2'\right\|}{\|\bar{m}_2\|}. \tag{14}$$

Since s_1, $\bar{s}_1$, $\|m_1(t)\|$, and $\left\|m_2'(t)\right\|$ are now known or measurable, (14) is used to solve for $\|\bar{m}_1(t)\|$ and $\|\bar{m}_2(t)\|$, which is used to recover $\bar{m}_1(t)$ and $\bar{m}_2(t)$ as

$$\bar{m}_1 = \frac{\|\bar{m}_1\|}{\|m_1\|} m_1, \ \bar{m}_2 = \frac{\|\bar{m}_2\|}{\|m_2\|} m_2.$$

Solutions for i_x, i_y, and $R(t)$ can now be determined from (8), (9), and (6), respectively. Since $\mathcal{F}_s(t)$ is attached to $\bar{m}_1(t)$, the translation is simply given by $x(t) = \bar{m}_1(t)$. Furthermore, since $\bar{m}_1(t)$ is determined, it is now possible to solve for the distance $d(t)$ as

$$d = n \cdot \bar{m}_1. \tag{15}$$

If the constant length $\bar{s}$ is not known, but $d(t)$ is known or estimated using known camera motions, then the fact that $d(t) = z_j(t)n^T(t)m_j(t)$ can be used to solve for the Euclidean coordinates of $\bar{m}_j(t)$ as

$$\bar{m}_j = \frac{dm_j}{n \cdot m_j}, \ \forall j \in \{1 \dots N\}. \tag{16}$$

Solving for the Euclidean coordinates of two points $\bar{m}_1(t)$ and $\bar{m}_2(t)$ allows $\bar{s}_1$ to be estimated, which can be used to solve for the frame $\mathcal{F}_s(t)$.

With d determined, and m_j, $\forall j \in \{1 \dots k\}$ measurable, the 3D coordinates in $\mathcal{F}_c(t)$ of all points $\bar{m}_j = [x_j, y_j, z_j]^T$, $\forall j \in \{1 \dots k\}$ is determined as

$$\bar{m}_j = z_j m_j = \frac{d}{n \cdot m_j} \bar{m}_j, \tag{17}$$

where

$$d = n \cdot \bar{m}_j = n \cdot m_j z_j \tag{18}$$

$$z_j = \frac{d}{n \cdot m_j}. \tag{19}$$

B. Visual Servoing to an Arbitrary Pose

A simple visual servo controller that utilizes the above method is now presented. The control objective is to move the camera to a goal pose by generating a continuous camera velocity $v(t) \in \mathbb{R}^6$. The strength of our method is that the camera is accurately regulated to a pose defined with respect to a planar object, i.e. teach by showing is not required. Other PBVS methods that use the Homography matrix (e.g. [9], [18]) are restricted to positioning the camera with respect to the camera poses $\mathcal{F}_c^*$ or $\mathcal{F}_c$. Using the method presented

here, there is no need for an a priori goal image or reference image. Other pose reconstruction methods to derive pose with respect to an object may require detailed structural information (e.g. [5], [6]).

Let R_d and x_d denote the desired, constant rotation and translation of the camera with respect to the static plane frame $\mathcal{F}_s$, such that the image points on the plane are visible to the camera at the desired pose. The camera is at a pose $\mathcal{F}_c^*$ that is currently unknown with respect to $\mathcal{F}_s$. By performing a small translation, the camera moves to a pose $\mathcal{F}_c(t_1)$ and the Homography $H(t_1)$ is computed and decomposed to $R(t_1)$, $\frac{x(t_1)}{d^*}$ and n^*.

With $n(t) = R(t)n^*$ known, the pose of $\mathcal{F}_s$ can be determined with respect to the current camera frame $\mathcal{F}_c(t)$. The rotation and translation between $\mathcal{F}_c(t)$ and $\mathcal{F}_s$ are denoted as $R_c(t)$ and $x_c(t)$ respectively. A constant, desired pose with respect to $\mathcal{F}_s$ is denoted as $\mathcal{F}_d$. The rotation error $R_e(t) \in SO(3)$ and translation error $x_e(t) \in \mathbb{R}^6$ between $\mathcal{F}_d$ and $\mathcal{F}_c(t)$ is given by

$$
\begin{align}
R_e &= R_d R_c^T \tag{20}\\
x_e &= -R_d R_c^T x_c + x_d. \tag{21}
\end{align}
$$

The rotation error matrix $R_e(t)$ is locally mapped to

$$
R_e(t) \rightarrow u(t)\theta(t) \in \mathbb{R}^3,
$$

where $\theta(t) \in \mathbb{R}$ is the rotation angle about the axis $u(t) \in \mathbb{R}^3$. The pose error vector is then defined as

$$
e = [x_e^T, \ u\theta^T]^T \in \mathbb{R}^6. \tag{22}
$$

The camera velocity is given by

$$
\xi = [v^T, \omega^T]^T \in \mathbb{R}^6, \tag{23}
$$

where $v(t) \in \mathbb{R}^3$ is the linear velocity and $\omega(t) \in \mathbb{R}^3$ is angular velocity. The time derivative of (22) is given as a function of camera velocity by

$$
\dot{e} = L\xi, \tag{24}
$$

where $L(t) \in \mathbb{R}^{6 \times 6}$ is a Jacobian-like matrix that maps camera velocity to pose error the derivative given as

$$
L = \begin{bmatrix} R_{vc} & 0_{3\times3} \\ 0_{3\times3} & R_{vc}L_\omega \end{bmatrix}. \tag{25}
$$

In (25), $R_{vc}(t) \in SO(3)$ is the rotation matrix from the frame in which $\xi(t)$ is measured to $\mathcal{F}_c(t)$. $R_{vc}(t)$ is identity if the camera frame and input velocity frame are the same. The Jacobian like matrix $L_\omega(t) \in \mathbb{R}^{3\times3}$ maps angular velocity to $\frac{d}{dt}(u\varphi)$. As shown in [9], $L_\omega(t)$ is given by

$$
L_\omega = I - \frac{\varphi}{2}u_\times + \left(1 - \frac{\text{sinc}(\varphi)}{\text{sinc}^2\left(\frac{\varphi}{2}\right)}\right)u_\times^2,
$$

where I is the 3×3 identity matrix, $u_\times(t) \in \mathbb{R}^3$ is the skew symmetric matrix form of the vector $u(t)$, $\text{sinc}(\varphi) = \frac{\sin(\varphi)}{\varphi}$, and $\text{sinc}(0) = 1$.

Based on (22) and (24), the linear and angular velocity input

$$
\xi = -\lambda L^{-1}e \tag{26}
$$
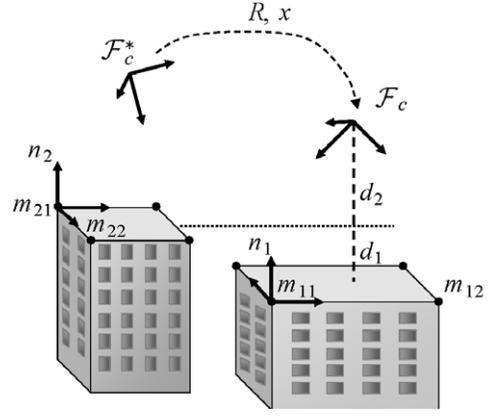
Fig. 2. Example of Multiple Planar Surfaces

can then be defined, where λ is a scalar gain. Combining (24) and (26) gives the closed-loop error derivative

$$
\dot{e} = -\lambda e. \tag{27}
$$

The closed-loop system in (26) can be shown to asymptotically stabilize the error to zero, thereby bringing the camera to the goal pose. A simulation this controller is presented in Section V-A.

IV. GEOMETRIC RECONSTRUCTION OF PIECEWISE PLANAR OBJECTS AND SCENES

A. Reconstructing the Geometry of Multiple Planes with Respect to the Camera

The previous development is now extended to the case of multiple planar patches and piecewise planar objects, given knowledge of only a single geometric length on a single static object in the scene. Consider a large sample of points $\mathcal{P}$ visible to the camera. These points are grouped into g sets of coplanar points $\mathcal{P}_h \subset \mathcal{P}$, $\forall h \in \{1 \dots g\}$, where all points in $\mathcal{P}_h$ lie in a plane π_h. The sets $\mathcal{P}_h$ may overlap, i.e., a point may be in more than one set.

Segmenting a set of points into coplanar sets is not a trivial task. The sets can be distinguished through human interaction, scene knowledge (e.g., multiple light objects on a dark background) or various automated methods [19], [20]. This example assumes that each set $\mathcal{P}_h$ is well conditioned in the sense that it contains no more than three collinear points, as illustrated in Fig. 2.

In the following development, the camera is assumed to undergo a rotation $R(t)$ and translation $x(t)$ from a reference frame $\mathcal{F}_c^*$ to a frame $\mathcal{F}_c(t)$. The points in each set $\mathcal{P}_h$ have coordinates

$$
\begin{align}
\bar{m}_{hj}^* &= [x_{hj}^*, y_{hj}^*, z_{hj}^*]^T, \\
\forall j &\in \{1 \dots N_h\}, \ \forall h \in \{1 \dots g\} \\
\bar{m}_{hj}(t) &= [x_{hj}, y_{hj}, z_{hj}]^T, \\
\forall j &\in \{1 \dots N_h\}, \ \forall h \in \{1 \dots g\}
\end{align}
$$

in the frames $\mathcal{F}_c^*$ and $\mathcal{F}_c(t)$, respectively. These points project to image points with normalized coordinates m_{hj}^*

and $m_{hj}(t)$, as described in Section II. Each set of points in the images are related by a homography $H_h(t)$ defined by

$$m_{hj} = \frac{z_{hj}^*}{z_{hj}} H_h m_{hj}^* \tag{28}$$

$$m_{hj} = \frac{z_{hj}^*}{z_{hj}} \left(R(t) + \frac{x(t)}{d_h^*} n_h^{*T} \right) m_{hj}^*. \tag{29}$$

Note that $R(t)$ and $x(t)$ are the same for all point sets $\mathcal{P}_h$, since all coordinate changes are due to the motion of the single camera. However, each plane π_h is different; therefore, each set of points will have different d_h^*, n_h^* and $H_h(t)$.

From m_{hj}^* and m_{hj}, it is possible to recover $H_h(t)$, n_h^*, $R(t)$, and $x_h(t) = \frac{x(t)}{d_h^*}$ for all $h \in \{1 \ldots g\}$. The subsequent development is based on the assumption that a single geometric length between two points in a single set is known. Without loss of generality, this length is assumed to be known in set $\mathcal{P}_1$. Given this geometric length, a reference frame $\mathcal{F}_{s1}$ is attached to plane π_1 and the development in Section III is used to solve for $R_{c1}(t)$, $x_{c1}(t)$, $d_1(t)$, d_1^* and all $\bar{m}_{1j}$, $\forall j \in \{1 \ldots N_1\}$. The translation $x(t)$ can then be recovered from d_1^* as

$$x(t) = d_1^* x_1(t).$$

Given $x(t)$, each d_h^*, $\forall h \in \{2 \ldots m\}$ is recovered from the scaled translations $x_h(t)$ as

$$d_h^* = \frac{x_h^T x}{\|x_h\|}.$$

Once each d_h^* has been determined, all $\bar{m}_{hj}^*$, $\forall j \in \{1 \ldots N_h\}$, $\forall h \in \{2 \ldots m\}$ can be recovered as in (16). From knowledge of $\bar{m}_{hj}^*$, a constant length $\bar{s}_h$ between two feature points can be estimated for each plane, and the frames $\mathcal{F}_{sh}$ can be attached to the plane π_h. Thus, (6)-(9) can be used to solve for R_{ch} and x_{ch}, $\forall h \in \{2 \ldots m\}$. Given the rotation and translation from each plane π_h to the reference camera frame $\mathcal{F}_c$, the rotation and translation between each planar patch can be recovered.

For a stationary camera viewing multiple moving planar objects, the analysis cannot be performed because there is not a common $R(t)$ or $x(t)$. If a geometric length is known on each object, then the analysis in Section III can be performed for each moving plane.

B. Visual Servoing a Piecewise Planar Object to an Arbitrary Pose with Respect to an Unknown Object

A simple visual servo controller that utilizes the development for multiple planar objects is now presented. The task is to move a piecewise planar object in the view of a camera to a goal pose with respect to a second, stationary planar object. If a single known length is available on the object, then by temporarily keeping that object static, that length can be used to calculate geometric knowledge of all planar objects in the scene. Once geometric knowledge of the scene is available, there is no need for the controlled object to remain static, and a positioning control task can be performed.

Assume a camera is at pose $\mathcal{F}_c^*$ and views a scene consisting of two planar objects π_1 and π_2, where π_1 can accept velocity inputs and π_2 is static. Assume further that the camera is capable of at least some small autonomous motion. After moving to an arbitrary pose $\mathcal{F}_c(t_1)$, the homographies $H_1(t)$ and $H_2(t)$ are determined. With $n_1(t) = R_1(t)n_1^*$ and $n_2(t) = R_1(t)n_2^*$ known, the pose of both planar objects are solved with respect to the camera frame $\mathcal{F}_c(t)$. The rotation and translation from $\mathcal{F}_{s1}$ to $\mathcal{F}_c(t)$, is denoted $R_{c1}(t)$ and $x_{c1}(t)$, respectively. Likewise, the rotation and translation from $\mathcal{F}_{s2}$ to $\mathcal{F}_c(t)$, are given by $R_{c2}(t)$ and $x_{c2}(t)$. For simplicity, assume that the camera does not move again after the reference frames have been assigned to the planes.

Define a constant, desired rotation and translation R_d and x_d with respect to $\mathcal{F}_{s2}$. The rotation $R_e(t)$ and translation $x_e(t)$ from the desired pose to the current camera pose $\mathcal{F}_c(t)$ is given by

$$R_e = (R_{c2} R_d)^T R_{c1}^T \tag{30}$$

$$x_e = -R_d(-R_{c1}^T R_{c2}^T x_{c2} + x_{c1}) + x_d. \tag{31}$$

The rotation matrix $R_e(t)$ is locally mapped to $R_e(t) \rightarrow u_e(t)\theta_e(t) \in \mathbb{R}^3$, where $\theta_e \in \mathbb{R}$ is the angle of rotation about the axis $u_e \in \mathbb{R}^3$. The pose error vector is then given by

$$e(t) = [x_e(t), \ u_e(t)\theta_e(t)] \in \mathbb{R}^6. \tag{32}$$

Similar to the discussion in Section V-A, a velocity vector for $\mathcal{F}_{s1}$ that will asymptotically stabilize the error in (32) is given by

$$\xi = -\lambda L^{-1} e, \tag{33}$$

where $L(t)$ is a matrix that maps the velocity of $\mathcal{F}_{s1}$ to the derivative of the pose error and λ is positive scalar gain. Simulations of such a controller are given in Section V-B.

V. EXPERIMENTAL RESULTS

A. Simulation of Eye In Hand Visual Servoing to an Arbitrary Pose

The control task described in Section III-B is simulated for a set of four coplanar points. The points are configured in a square, but only the distance between two points is assumed to be known. There is perfect camera calibration, and quantization noise is added by rounding the image feature coordinates to the nearest integer. The desired camera pose is $[x_d^T, u_d^T \theta_d] = [0, 0, -1.2, 0, 0, \pi]^T$ with respect to a frame $\mathcal{F}_s$ attached to one of the feature points. The initial pose error is $[x_e^T(0), u_e^T(0)\theta_e(0)] = [0.6721, \ 0.7489, \ 0.3471, \ 0.4092, \ -0.5855, \ -0.1970]$, and the developed method estimates the initial pose error to be $[x_e^T(0), u_e^T(0)\theta_e(0)] = [0.77610, \ 0.7378, \ 0.3173, \ 0.4086, \ -0.5846, \ -0.1967]$. Visual servoing is performed to regulate the pose error. The pose error over time is given in Fig. 4. The trajectory of the feature points in the image is given in Fig. 3.

B. Simulation of Vehicle Regulation with Respect to an Unknown Object

A simulation of the task given in Section IV-B is presented in this section. The objective is to move a controllable planar
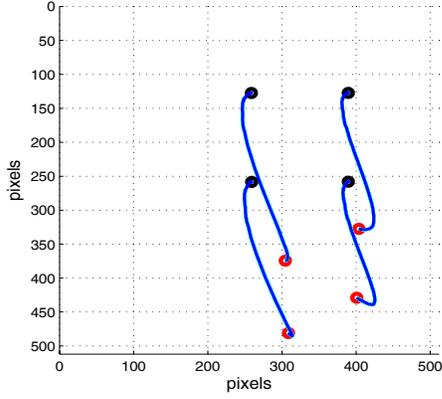
Fig. 3.   Feature point trajectories

Fig. 4.   Pose error over time

Fig. 5.   Trajectory of controlled patch
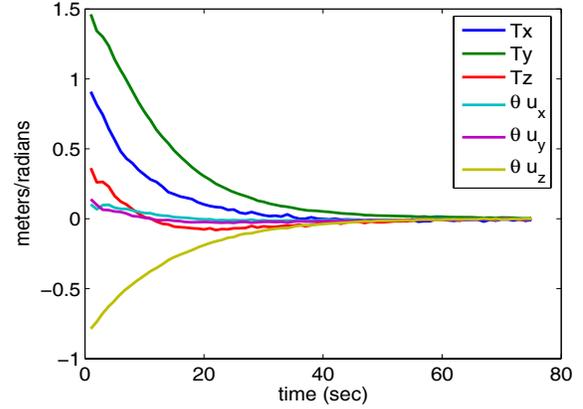
Fig. 6.   Pose error over time for controlled patch

object to a pose $[x_d^T,\ u_d^T\theta_d] = [0, 0, .5, 0, 0, \frac{\pi}{2}]$ with respect to a static planar object. Both planar objects are rectangular, but only a single known length on the controllable object is given. The initial and final poses, along with the choices of $\bar{m}_{11}, \bar{m}_{12}, \bar{m}_{21}$ and $\bar{m}_{22}$ were chosen such that this task is similar to positioning a known vehicle in front of a previously unknown building. The path of the planar object as seen by the camera is shown in Fig. 5. The pose error of $\mathcal{F}_{s1}$ with respect to $\mathcal{F}_d$ is shown in Fig. 6.

C. Experiments of Pose Estimation of a Single Object

In conjunction with the Center for Intelligent Machines and Robotics at the University of Florida, we are in the initial stages of implementing the method presented in the paper in a vehicle following (i.e., platooning) applications. A leader vehicle is fixed with a known target, and a follower vehicle carries an on-board camera. By tracking the target points on the leader vehicle, the pose estimation method presented in this paper and the vision-based velocity estimation method presented in [15] are used to estimate relative pose and velocity of the leader vehicle with respect to follower vehicle. The follower vehicle uses a predefined optimization function to trace its path in the environment. The relative pose estimation allows the follower to maintain a fixed distance

between the leader and follower vehicle. The throttle and brakes can be controlled with a PD controller, using a relative pose and velocity estimation data, to avoid collision with leader vehicle or losing track of leader vehicle in the field. Early experiments demonstrate the effectiveness of the vision based estimation in this scheme. Field tests of the complete platooning system will be conducted in the coming weeks.

An experiment using a moving vehicle in Fig. 7 is performed to demonstrate the pose estimation method as used in the platooning application. Four bright LED arrays were fixed to the back of a truck to facilitate simple image segmentation, where the centroid of each detected array provides four feature points used to construct the Homography matrix. Each of the four centroids is indicated in Fig. 7 by a cross and a number. The truck was equipped with a differential GPS unit to provide a reference to compare the pose estimation. The road was marked at approximately 20 feet (6.1 m) intervals and the car was driven forward and stopped approximately every 20 feet.

The results of the experiment are seen in Fig. 8. The expected periodically increasing step function along the camera frame z direction (i.e., the optical axis) is evident. Furthermore, the change in pose estimate agrees closely to the GPS Northing measurement. There is also a small periodic step increase estimated in the camera x direction. The estimate degrades as the distance to the vehicle increases. This is primarily due to sensor noise. As the car moves

Fig. 7. A processed video frame from the pose estimation experiment.
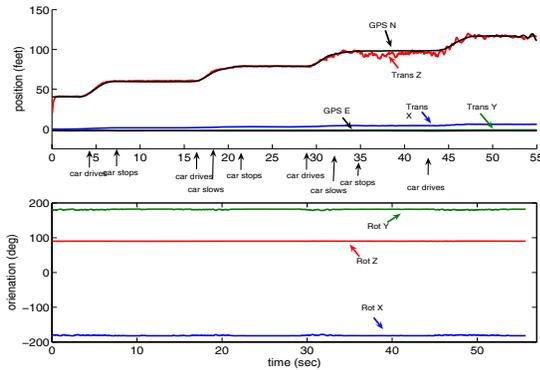
Fig. 8. Results of experiment to estimate the pose of a moving vehicle.

farther from the camera, the perceived lights become dimmer and it is harder to extract the centroids. This increases the effects of pixilation (i.e., quantization noise).

There are currently several avenues of development on the vision system. The first is the introduction of brighter LED arrays, which will mitigate the problem of feature noise at large distances. New set of LED's are visible at a distance up to 100 meters. Additional LED targets will be affixed to the sides of the lead vehicle to maintain target tracking during turns or aggressive maneuvers. The next experiment will involve 5 targets at the back, and 4 each on left and right sides of the leader vehicle. Five targets on the back plane will be used to distinguish back plane from side planes in an image. Efforts are being made to replace artificial targets with natural features like sharp corners.

VI. Conclusions and Future Work

In this paper, a method for geometric reconstruction is presented that uses a Euclidean homography between two images of coplanar points and a single known geometric length. This method is designed for use in PBVS, and offers the ability to control the pose of the camera or a body with respect to a planar object in the scene. This distinguishes our method from other PBVS methods that use the homography but are use teach by showing techniques that are only able to position the camera or object with respect to the pose in a goal image. The presented method is also distinguished from pose estimation methods that require a detailed CAD model to solve for the camera's pose with respect to an object, since only a single known length is necessary.

The method developed in this paper is also suitable for numerous vision-based estimation methods. Since the relative locations of all planar surfaces and their respective feature points is known through this method, it remains possible to estimate the location of features if they leave the image or are occluded. We explore this application with regards to a rotating polyhedron in [16]. This method has been verified in simulations of visual servoing tasks and experimental pose estimation of moving ground vehicle. Future work will focus on using the method for more complicated control tasks, including closed loop control of aerial and ground vehicles.

References

[1] P. Martinet, J. Gallice, and D. Khadraoui, "Vision based control law using 3D visual features," in *Proc. WAC 96*, vol. 3, pp. 497–502, 1996.
[2] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 651–670, Oct. 1996.
[3] G. S. B. W. J. Wilson, C. C. W. Hulls, "Relative end-effector control using cartesian position based visual servoing," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 684–696, 1996.
[4] N. Daucher, M. Dhome, J. Lapresté, and G. Rives, "Speed command a robotic system by monocular pose estimate," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 55–43, 1997.
[5] D. DeMenthon and L. S. Davis, "Model-based object pose in 25 lines code," in *European Conf. on Computer Vision*, pp. 335–343, 1992.
[6] L. Quan and Z.-D. Lan, "Linear n-point camera pose determination," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 8, pp. 774–780, 1999.
[7] O. D. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *Int. J. Pattern Recog. and Artificial Intell.*, vol. 2, no. 3, pp. 485–508, 1988.
[8] Y. Ma, S. Soatto, J. Kosecká, and S. Sastry, *An Invitation to 3-D Vision*. Springer, 2004.
[9] E. Malis, F. Chaumette, and S. Boudet, "2-1/2D visual servoing," *IEEE Trans. Robot. Automat.*, vol. 15, no. 2, pp. 238–250, 1999.
[10] C. Taylor and J. Ostrowski, "Robust vision-based pose control," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2734–2740, 2000.
[11] Y. Fang, W. E. Dixon, D. M. Dawson, and P. Chawda, "Homography-based visual servo regulation of mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 35, no. 5, pp. 1041–1050, 2005.
[12] 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.
[13] G. Silveira, E. Malis, and P. Rives, "Visual servoing over unknown, unstructured, large-scale scenes," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 4142– 4147, 2006.
[14] F. C. C. Collowet, "Positioning a camera with respect to planar objects unknown shape by coupling 2-d visual servoing and 3-d estimations," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 322–333, June 2002.
[15] V. Chitrakaran, D. M. Dawson, W. E. Dixon, and J. Chen, "Identification a moving object´s velocity with a fixed camera," *Automatica*, vol. 41, no. 3, pp. 553–562, 2005.
[16] K. Dupree, N. Gans, W. Mackunis, and W. Dixon, "Euclidean feature tracking for a rotating satellite," in *Proc. American Control Conference*, pp. 3874–3879, 2007.
[17] Z. Zhang and A. Hanson, "3D reconstruction based on homography mapping," in *Proc. ARPA Image Understanding Workshop Palm Springs CA*, 1996.
[18] Y. Fang, D. Dawson, W. Dixon, and M. de Queiroz, "Homography-based visual servoing wheeled mobile robots," in *Proc. IEEE Conf. on Decision and Control*, pp. 2866–2871, 2002.
[19] C. Baillard and A. Zisserman, "Automatic reconstruction planar models from multiple views," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 559–565, 1999.
[20] K. Okada, S. Kagami, M. Inaba, and H. Inoue, "Plane segment finder: algorithm, implementation and applications," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2120–2125, 2001.