In Proc. of the 36th Conf. on Decision and Contr., Dec. 1997.

TASK RE-ENCODING IN VISION-BASED CONTROL SYSTEMS

Wen-Chung Chang
wchang@cc.ndhu.edu.tw

João P. Hespanha
hespanha@yale.edu

A. S. Morse
morse@sysc.eng.yale.edu

Gregory D. Hager
hager@cs.yale.edu

Center for Computational Vision and Control
Yale University,
New Haven, CT 06520

ABSTRACT

Feedback control systems employing video cameras as sensors have been studied in the robotics community for many years. An especially interesting feature of such systems is that *both* the process output {e.g., the position of the robot in its workspace} and the reference set-point {e.g., visually determined target} are typically observed through the *same* sensors {i.e., cameras}. Because of this unusual architectural feature, it is sometimes possible to achieve *precise* positioning {in the absence of measurement noise}, despite sensor/actuator and process model imprecision, just as it is in the case of a conventional set-point control system with a perfect loop-integrator and precise output and exogenous reference sensing. But in contrast to a set-point control system where what to choose for an error is usually clear, in vision-based systems there are many choices for errors, each with different attributes. The aim of this paper is to discuss these issues in a fairly general setting and to provide concrete examples to illustrate the concepts involved in geometrical terms.

1. INTRODUCTION

Feedback control systems employing video cameras as sensors have been studied in the robotics community for many years {c.f. a recent tutorial on visual servoing [1]}. An especially interesting feature of such systems is that *both* the process output {e.g., the position of the robot in its workspace} and the reference set-point {e.g., visually determined target} are typically observed through the *same* sensors {i.e., cameras}. Because of this unusual architectural feature, it is sometimes possible to achieve *precise* positioning {in the absence of measurement noise}, despite sensor/actuator and process model imprecision [2, 3, 4, 5], just as it is in the case of a conventional set-point control system with a perfect loop-integrator and precise output and exogenous reference sensing. But in contrast to a set-point control system where what to choose for an error is usually clear, in vision-based systems there are many choices for errors, each with different attributes.

This research was supported by the Air Force Office of Scientific Research, the Army Research Office, and the National Science Foundation.

Some of the observations just made are implicit in work extending back more than 10 years [6]. Some of these issues are touched upon in [7] and in the monograph [8]. The concept of task encoding is discussed briefly in the Doctoral Thesis of [9] which contains further references.

The aim of this paper is to discuss how to achieve precise positioning, despite sensor/actuator and process model imprecision, in a fairly general setting. In Sections 2 and 3 we briefly describe the vision/control system of interest and we define a generalized positioning task. The task can not be accomplished as stated because it depends on data which can not be precisely modeled or measured. In Section 4 we discuss three different re-encodings of this task and give conditions under which each can achieve precise positioning in the face of camera modeling errors. Three geometrical examples illustrating these findings are then given in Section 5. A full length version of this paper is available [10].

2. THE SYSTEM

This paper considers the problem of controlling the position of a rigid robot in a prescribed *workspace* $\mathcal{W} \subset \mathbb{R}^3$ using data observed by a two-camera vision system. The observed data consists of the robot position in $\mathcal{W}$ as well as various geometrical features of interest which appear in the cameras' fields of view $\mathcal{V} \subset \mathbb{R}^3$. Invariably $\mathcal{W} \subset \mathcal{V}$.

The robot is assumed to admit a simple kinematic model of the form

$$\dot{x} = u \tag{1}$$

where x is a state vector whose components are the Cartesian coordinates of the robot in $\mathcal{W}$, and u is a control vector taking values in $\mathbb{R}^3$. Visually identifiable landmarks are modeled as subsets of the cameras' field of view $\mathcal{V}$ {e.g., a line segment, a circle, etc.} and are referred to as *features*.

The robot's position in $\mathcal{W}$ as well as various features of interest in $\mathcal{V}$ are seen in an *image space* $\mathcal{Y} \triangleq \mathbb{R}^2 \oplus \mathbb{R}^2$ through a fixed but imprecisely known readout function or *two-camera model* $C_{\text{actual}} : \mathcal{V} \to \mathcal{Y}$ which describes a two-camera vision system. Thus the robot's position in image space is a measured output y related to x by the formula

$$y = C_{\text{actual}}(x). \tag{2}$$

Correspondingly, a specific feature {i.e. subset} f in $\mathcal{V}$ is seen in $\mathcal{Y}$ as[1]

$$z = C_{\text{actual}}(f). \tag{3}$$

Note that because f is a subset of $\mathcal{V}$, it is also an element of the power set[2] of $\mathcal{V}$ which we henceforth denote by $\bar{\mathcal{V}}$. Similarly z is an element of the power set of $\mathcal{Y}$, which we write as $\bar{\mathcal{Y}}$. Equations (1)–(3) thus model the robot/vision system of interest.

Generally speaking one would need C_{actual} to be at least injective in order for the vision subsystem to provide the requisite information for positioning. On the other hand, one would never expect to know precisely what C_{actual} is. Accordingly, we will assume that C_{actual} is a fixed but unknown element of a prescribed set $\mathcal{C}$ of injective functions each mapping $\mathcal{V}$ into $\mathcal{Y}$. For the present we will place no constraints on $\mathcal{C}$. It might for example be a finite set, a neighborhood of C_{actual} in a metric space, a finite collection of disjoint neighborhoods or something else.

For each two-camera model $C \in \mathcal{C}$, there is a naturally induced injective function $\bar{C}$ from $\bar{\mathcal{V}}$ to $\bar{\mathcal{Y}}$ defined by the rule $\bar{x} \longmapsto C(\bar{x})$. The $\bar{C}$ are injective because the C are. In this notation, (3) can also be written as

$$z = \bar{C}_{\text{actual}}(f). \tag{4}$$

In the sequel, (3) is preferred, but where it may lead to confusion (4) is used instead.

3. THE TASK

A typical positioning problem for the type of system we are considering consists of driving a robot to a 'target' {or set} in $\mathcal{W}$ determined by a list of observed features in $\mathcal{V}$. In the sequel we will formalize this problem in general terms, deferring examples to §5.

In practice, target sets are determined not by just one observed feature, but rather by an ordered set of n simultaneously observed features $\{f_1, f_2, \ldots, f_n\}$ where each f_i is a member of some specific class $\mathcal{F}_i$ of like or similar objects. For example, $\mathcal{F}_1$ might be a set of line segments, $\mathcal{F}_2$ a set of cubes, etc. In the sequel $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n$ are n such feature classes and $\mathcal{F}$ is a given *composite feature space* consisting of all lists $\{f_1, f_2, \ldots, f_n\}$ of interest, where each $f_i \in \mathcal{F}_i$. Thus $\mathcal{F}$ is a pre-specified subset of $\mathcal{F}_1 \times \mathcal{F}_2 \times \cdots \times \mathcal{F}_n$ which is in turn a subset of $\tilde{\mathcal{V}} \triangleq \underbrace{\bar{\mathcal{V}} \times \bar{\mathcal{V}} \times \cdots \times \bar{\mathcal{V}}}_{n \text{ times}}$.

The relationship between observed lists of features {e.g., two features each consisting of a single point} and targets {e.g., the line passing through the two points} is typically characterized by a given function $T : \mathcal{F} \to \bar{\mathcal{V}}$ called a *target map*. Each list of features f in $\mathcal{F}$ specifies a possible target set $T(f) \subset \mathcal{W}$ to which we might like to move the robot. In other words, for a given list of features $f = \{f_1, f_2, \ldots, f_n\} \in \mathcal{F}$, the *task* to be accomplished is to cause the robot state to satisfy

$$\boxed{x \in T(f).} \tag{5}$$

The control problem is thus to develop a causal algorithm which accomplishes (5), using only sensed data $C_{\text{actual}}(x)$ and $\{C_{\text{actual}}(f_1), C_{\text{actual}}(f_2), \ldots, C_{\text{actual}}(f_n)\}$, subject to the assumption that C_{actual} is in $\mathcal{C}$ but is otherwise unknown.

Since C_{actual} is not presumed known with precision, neither x nor the features f_i can be determined without error. Therefore the desired task (5) may not be accomplished as it stands. The objective of this paper is to discuss several alternative tasks which can be implemented without knowing C_{actual} and which, if accomplished, imply that the original task (5) is accomplished as well.

4. RE-ENCODING

In the sequel are discussed three distinct ways to re-encode the task defined by (5) into tasks depending only on known information. In no case is C_{actual} assumed to be known. Notwithstanding this, the re-encodings are crafted with the goal of achieving *precise* positioning; i.e., satisfying (5). Examples which serve to illustrate and clarify the three approaches are discussed in §5.

4.1. Cartesian-Based Approach

The widely used Cartesian-based approach [2, 11, 12, 13, 14, 5] is motivated by the intuitive idea of "certainty equivalence." In the present context, certainty equivalence advocates that one should use estimates of x and f to accomplish task (5), with the understanding that these estimates are to be viewed as correct even though they may not be. The construction of such estimates starts with the selection {by some means} of a two-camera model C_{estim} in $\mathcal{C}$ which, in the context of certainty equivalence, is considered to be an approximate model of C_{actual}. Assume that such a C_{estim} has been chosen and {because of C_{estim}'s injectivity} that a specific left-inverse C^{-1}_{estim} has been correspondingly computed. In view of (2), it is natural to think of

$$\widehat{x} \triangleq C^{-1}_{\text{estim}}(y) \tag{6}$$

as an estimate of x.

[1] With $\mathcal{A}$ and $\mathcal{B}$ sets, $\mathcal{C} \subset \mathcal{A}$, and $f : \mathcal{A} \to \mathcal{B}$, we write $f(\mathcal{C})$ for the f-image of $\mathcal{C}$ which is defined by $\{f(a) \mid a \in \mathcal{C}\}$.

[2] The power set of a set $\mathcal{A}$ is the set of all subsets of $\mathcal{A}$.

To develop a corresponding estimate of f, we need first an expression relating f to what can be observed, namely the list

$$\{C_{\text{actual}}(f_1), C_{\text{actual}}(f_2), \ldots, C_{\text{actual}}(f_n)\}$$

Thus, for each $C \in \mathcal{C}$, let $\tilde{C}$ denote the map $\tilde{C} : \mathcal{F} \rightarrow \tilde{\mathcal{Y}} \triangleq \underbrace{\bar{\mathcal{Y}} \times \bar{\mathcal{Y}} \times \cdots \times \bar{\mathcal{Y}}}_{n \text{ times}}$ defined by the rule

$$\{f_1, f_2, \ldots, f_n\} \longmapsto \{C(f_1), C(f_2), \ldots, C(f_n)\}.$$

The preceding enables us to write

$$z = \tilde{C}_{\text{actual}}(f) \tag{7}$$

where z is the measured list of observed features

$$z \triangleq \{C_{\text{actual}}(f_1), C_{\text{actual}}(f_2), \ldots, C_{\text{actual}}(f_n)\}.$$

Since for purposes of certainty equivalence, C_{estim} is our current estimate of C_{actual}, it makes sense to estimate f using a left inverse of $\tilde{C}_{\text{estim}}$. Such left inverses exist because the $\tilde{C}$ are injective. This, in turn, is a direct consequence of the assumed injectivity of the C. In view of (7), it is natural to define

$$\widehat{f} \triangleq \tilde{C}_{\text{estim}}^{-1}(z) \tag{8}$$

as the estimate of f to be considered.

Since $C_{\text{estim}}^{-1}(y)$ and $\tilde{C}_{\text{estim}}^{-1}(z)$ are estimates of x and f respectively, in accordance with certainty equivalence, to accomplish the task (5) one should seek a control u to accomplish the re-encoded task

$$\boxed{C_{\text{estim}}^{-1}(y) \in T\big(\tilde{C}_{\text{estim}}^{-1}(z)\big).} \tag{9}$$

The question to be answered is this: When does accomplishing the re-encoded task (9) imply that the original task (5) has been accomplished as well? The following lemma gives a partial answer.

Lemma 1 *Let $C_{\text{estim}} \in \mathcal{C}$ be given and suppose that C_{estim}^{-1} is a fixed left inverse of C_{estim}. Let $\bar{C}_{\text{estim}}^{-1} : \bar{\mathcal{Y}} \rightarrow \bar{\mathcal{V}}$ denote the left inverse of $\bar{C}_{\text{estim}}$ defined by the rule $\bar{y} \longmapsto C_{\text{estim}}^{-1}(\bar{y})$. For each $x \in \mathcal{W}$ and each $f \in \mathcal{F}$ such that $T(f) \subset \mathcal{W}$, (9) implies (5) provided the composed map*

$$C_{\text{estim}}^{-1} \circ C_{\text{actual}}$$

is injective on $\mathcal{W}$ and

$$[\bar{C}_{\text{estim}}^{-1} \circ \bar{C}_{\text{actual}} \circ T](f) = [T \circ \tilde{C}_{\text{estim}}^{-1} \circ \tilde{C}_{\text{actual}}](f). \tag{10}$$

In the sequel, we say that a pair $\{C, C^{-1}\}$, with $C \in \mathcal{C}$ and C^{-1} a left inverse of C, is an *admissible bi-model* if[3] $[C^{-1} \circ C_{\text{actual}}]|_{\mathcal{W}}$ is an injective function. Note that because $[C_{\text{actual}}^{-1} \circ C_{\text{actual}}]|_{\mathcal{W}}$ is the identity on $\mathcal{W}$ and therefore injective, $\{C_{\text{actual}},\ C_{\text{actual}}^{-1}\}$ is an admissible bi-model. As regards robustness, this means that under suitable technical conditions, if $\mathcal{C}$ is a sufficiently small open neighborhood about C_{actual}, then each $C \in \mathcal{C}$ would have a left inverse which makes $\{C,\ C^{-1}\}$ an admissible bi-model.

In most applications of interest, one would like to be sure that accomplishment of the re-encoded task (9) implies that the original task (5) has been accomplished as well, no matter what list of features $f \in \mathcal{F}$ might be. In view of Lemma 1, one way to insure that this is so is to require $\{C_{\text{estim}}, C_{\text{estim}}^{-1}\}$ to be a bi-model and that (10) hold for all $f \in \mathcal{F}$; i.e.,

$$\bar{C}_{\text{estim}}^{-1} \circ \bar{C}_{\text{actual}} \circ T = T \circ \tilde{C}_{\text{estim}}^{-1} \circ \tilde{C}_{\text{actual}}. \tag{11}$$

When (11) holds we say that $C_{\text{estim}}^{-1} \circ C_{\text{actual}}$ and T *commute.* There is however no reason to expect that this commuting property will hold; typically it does not unless $n = 1$ and the target map T is the identity[4]. One way to deal with this problem is discussed next.

4.2. Modified Cartesian-Based Approach

The modified Cartesian-based approach, which is apparently a new idea, is motivated by a desire to avoid the stringent commuting requirement (11). The starting point for the approach is the requirement that the target map T be "invariant" on $\mathcal{C}$. Invariance is defined as follows.

A target map T is said to be *invariant* on $\mathcal{C}$ if there exists a function $\tilde{T} : \tilde{\mathcal{Y}} \rightarrow \bar{\mathcal{Y}}$ such that

$$\bar{C} \circ T = \tilde{T} \circ \tilde{C}, \qquad \forall C \in \mathcal{C}. \tag{12}$$

$\tilde{T}$ can be seen as a function which maps the two-camera image of a list of features into the two-camera image of the corresponding target set. As we will see, for example for perspective projection camera models, demanding that the invariance property holds turns out to be much less severe than requiring the commuting property (11) to hold {c.f. §5}.

Remark 1 It is a simple matter to verify that for each $C \in \mathcal{C}$, the equation

$$\bar{C} \circ T = \tilde{T}_C \circ \tilde{C} \tag{13}$$

holds with $\tilde{T}_C = \bar{C} \circ T \circ \tilde{C}^{-1}$. Thus T is invariant on $\mathcal{C}$ if it is possible to find a single function $\tilde{T}_C$, not depending on C, for which (13) holds for all $C \in \mathcal{C}$. ♠

To proceed, assume that a function $\tilde{T}$ has been found for which the invariance property holds. As in the Cartesian-based approach, to re-encode one also needs to select a C_{estim} in $\mathcal{C}$ which in the context of certainty equivalence,

[3] With $\mathcal{A}$ and $\mathcal{B}$ sets, $\mathcal{C} \subset \mathcal{A}$, and $f : \mathcal{A} \rightarrow \mathcal{B}$, we write $f|_{\mathcal{C}}$ for the *restricted* map $\mathcal{C} \rightarrow f(\mathcal{C}) : c \longmapsto f(c)$.

[4] This corresponds to moving the robot to the feature itself.

is considered to be an estimate of C_{actual}. We assume that such a C_{estim} has been chosen and that C_{estim}^{-1} is a fixed left inverse of C_{estim}. In contrast to the Cartesian-based approach which seeks to accomplish the task defined by (9), the modified Cartesian approach seeks to accomplish the task

$$\boxed{C_{\text{estim}}^{-1}(y) \in C_{\text{estim}}^{-1}\big(\tilde{T}(z)\big)} \tag{14}$$

instead. The following lemma provides justification for this approach.

Lemma 2 *Let $C_{\text{estim}} \in \mathcal{C}$ be given and suppose that C_{estim}^{-1} is a fixed left inverse of C_{estim}. Suppose in addition that $\tilde{T}$ is a fixed function for which the invariance property (12) holds. For each $x \in \mathcal{W}$ and each $f \in \mathcal{F}$ such that $T(f) \subset \mathcal{W}$, (14) implies (5) provided $\{C_{\text{estim}},\ C_{\text{estim}}^{-1}\}$ is an admissible bi-model.*

It should be emphasized that Lemma 2 does not require C_{estim} to be "close" to C_{actual} in any particular sense.

4.3. Image-Based Approach

As with the modified Cartesian-based approach, the image-based approach also requires that the target map T be invariant on $\mathcal{C}$. In other words there must be a function $\tilde{T} : \tilde{\mathcal{Y}} \to \bar{\mathcal{Y}}$ such that

$$\bar{C} \circ T = \tilde{T} \circ \tilde{C}, \qquad \forall C \in \mathcal{C}. \tag{15}$$

To proceed, assume that the invariance property is satisfied by some computable function $\tilde{T}$. In contrast to the modified Cartesian approach which seeks to accomplish the task defined by (14), the image-based approach seeks to accomplish

$$\boxed{y \in \tilde{T}(z)} \tag{16}$$

instead. The approach is justified by the following lemma.

Lemma 3 *Suppose that $\tilde{T}$ is a fixed function for which the invariance property (12) holds. For each $x \in \mathcal{W}$ and each $f \in \mathcal{F}$, (16) implies (5).*

Accomplishing the task defined by (16) clearly causes the task defined by (14) to be accomplished as well. But, as opposed to both Cartesian-based approaches, the image-based approach does not require the selection of a candidate two-camera model C_{estim} in $\mathcal{C}$ to serve as an estimate of C_{actual}. However, in practice, designing a controller that accomplishes (16) will typically require such an estimate. Discussions of image-based approaches can be found in [6, 15, 16, 17, 18, 19, 3, 4].

5. EXAMPLES

Several tasks are introduced below. For each, we determine whether or not the commuting (11) and invariance (12) properties hold. Conclusions regarding applicability and effectiveness of the three approaches are drawn for each task. We assume that a family of admissible two-camera models $\mathcal{C}$ is given such that each C in $\mathcal{C}$ is the standard two-camera perspective projection model [20] of the form

$$C(x) = \begin{bmatrix} f_1 \frac{e_1' R_1(x-c_1)}{e_3' R_1(x-c_1)} \\ f_1 \frac{e_2' R_1(x-c_1)}{e_3' R_1(x-c_1)} \\ f_2 \frac{e_1' R_2(x-c_2)}{e_3' R_2(x-c_2)} \\ f_2 \frac{e_2' R_2(x-c_2)}{e_3' R_2(x-c_2)} \end{bmatrix}$$

where R_1, R_2 are 3 by 3 orthogonal matrices with determinant equal to 1, c_1, c_2 are 3-vectors, and e_i $\{i = 1, 2, 3\}$ is the ith column of the 3 by 3 identity matrix.

Example 1 (Point-to-Point Positioning) The task is to move the robot x to a point r in the robot workspace $\mathcal{W}$ as shown in Figure 1. The point r is the only feature that is

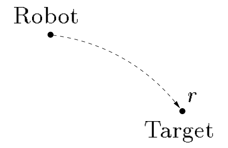

Figure 1: Point-to-Point Positioning

seen by the cameras. In this example, n is equal to 1 which means that $\tilde{\mathcal{V}} = \bar{\mathcal{V}}$ and $\tilde{\mathcal{Y}} = \bar{\mathcal{Y}}$. The composite feature space $\mathcal{F}$ is just the family of sets with a single element which is in $\mathcal{W}$, i.e.,

$$\mathcal{F} = \{\{r\} \,|\, r \in \mathcal{W}\}.$$

The target map $T : \mathcal{F} \to \bar{\mathcal{V}}$ is defined by the rule $\{r\} \longmapsto \{r\}$. This map is invariant on $\mathcal{C}$ since one can take $\tilde{T}$ to be the identity on $\tilde{\mathcal{Y}}$ {c.f. (12)}. Moreover, for any admissible bi-model $\{C_{\text{estim}}, C_{\text{estim}}^{-1}\}$, it is also true that T and $C_{\text{estim}}^{-1} \circ C_{\text{actual}}$ commute {c.f. (11)}. Therefore, all three approaches assure precise positioning.

Example 2 (Point-to-Point Positioning with Multiple Features) The task is to move the robot x to the point of intersection in $\mathcal{W}$ of two lines each one defined by two feature points, e.g., $\{r_1, r_3\}$ and $\{r_2, r_4\}$ as shown in Figure 2. The four feature points $r_1, r_2, r_3, r_4 \in \mathcal{V}$ are sensed by the cameras. In this example n is equal to 4 which means that $\tilde{\mathcal{V}} = \bar{\mathcal{V}} \times \bar{\mathcal{V}} \times \bar{\mathcal{V}} \times \bar{\mathcal{V}}$ and $\tilde{\mathcal{Y}} = \bar{\mathcal{Y}} \times \bar{\mathcal{Y}} \times \bar{\mathcal{Y}} \times \bar{\mathcal{Y}}$. The composite feature space $\mathcal{F}$ is defined by

$$\mathcal{F} = \Big\{ \{\{r_1\}, \{r_2\}, \{r_3\}, \{r_4\}\} \,\Big|\, r_1, r_2, r_3, r_4 \in \mathcal{V} \text{ and the lines defined by } \{r_1, r_3\} \text{ and } \{r_2, r_4\} \text{ intersect at a single point in } \mathcal{W} \Big\}.$$

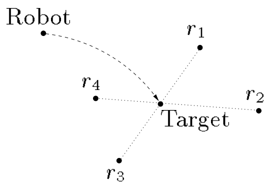

Figure 2: Point-to-Point Positioning with Multiple Features

The target map $T : \mathcal{F} \to \bar{\mathcal{V}}$ is defined by the rule

$$\{\{r_1\}, \{r_2\}, \{r_3\}, \{r_4\}\} \mapsto \{r_1 + \alpha_1 (r_3 - r_1) \mid r_1 + \alpha_1 (r_3 - r_1) = r_2 + \alpha_2 (r_4 - r_2)^5, \text{ for some } \alpha_1, \alpha_2 \in \mathbb{R}\}.$$

Defining $\tilde{T} : \tilde{\mathcal{Y}} \to \bar{\mathcal{Y}}$ by the rule

$$\begin{aligned} &\{\{h_1\}, \{h_2\}, \{h_3\}, \{h_4\}\} \mapsto \\ &\mapsto \{h_1 + \operatorname{diag}\{\beta_1, \beta_1, \beta_3, \beta_3\} (h_3 - h_1) \mid \\ &h_1 + \operatorname{diag}\{\beta_1, \beta_1, \beta_3, \beta_3\} (h_3 - h_1) = \\ &= h_2 + \operatorname{diag}\{\beta_2, \beta_2, \beta_4, \beta_4\} (h_4 - h_2), \\ &\qquad \text{for some } \beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{R}\}, \end{aligned}$$

which maps the four points $h_1, h_2, h_3, h_4 \in \mathcal{Y}$ to the intersection of two lines defined by $\{h_1, h_3\}$ and $\{h_2, h_4\}$, one can show that T is invariant on $\mathcal{C}$. This is a consequence of the fact that collinearity is preserved by perspective projection, therefore the image of the intersection of two lines is the intersection of the images of the two lines. However, picking an arbitrary bi-model $\{C_{\text{estim}}, C_{\text{estim}}^{-1}\}$, in general, T and $C_{\text{estim}}^{-1} \circ C_{\text{actual}}$ do not commute. Hence, the Cartesian-based approach can not guarantee precise positioning whereas both the modified Cartesian-based and the image-based approaches can assure precise positioning.

Example 3 (Point-to-Point Positioning with Set Feature) The task is to move the robot x to the center of a circle f in the cameras' field of view $\mathcal{V}$ as shown in Figure 3. It is assumed that the center of f lies in the robot workspace $\mathcal{W}$. The circle f is the only feature sensed by the cameras.

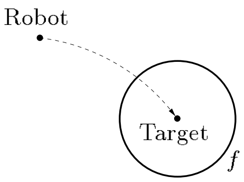

Figure 3: Point-to-Point Positioning with Set Feature

In this example, n is equal to 1 which means that $\tilde{\mathcal{V}} = \bar{\mathcal{V}}$ and $\tilde{\mathcal{Y}} = \bar{\mathcal{Y}}$. Let $\varphi(r_0, m, a)$ denote the circle with radius a, centered at $r_0 \in \mathbb{R}^3$, that lies in the plane orthogonal to $m \in \mathbb{R}^3$; i.e.,

$$\varphi(r_0, m, a) \triangleq \{r \in \mathbb{R}^3 \mid (r - r_0)'m = 0, \ \|r - r_0\| = a\}.$$

The composite feature space is the family of circles in $\mathcal{V}$ whose center is in $\mathcal{W}$, i.e.,

$$\mathcal{F} = \{\varphi(r_0, m, a) \mid r_0 \in \mathcal{W}, m \in \mathbb{R}^3, a \in \mathbb{R}^+, \text{ such that } \varphi(r_0, m, a) \subset \mathcal{V}\}.$$

The target map $T : \mathcal{F} \to \bar{\mathcal{V}}$ is defined by the rule

$$\varphi(r_0, m, a) \longmapsto \{r_0\}.$$

In general, the target map T is not invariant on $\mathcal{C}$, hence both the modified Cartesian-based and the image-based approaches do not apply. Moreover, picking an arbitrary bi-model $\{C_{\text{estim}}, C_{\text{estim}}^{-1}\}$, in general, T and $C_{\text{estim}}^{-1} \circ C_{\text{actual}}$ do not commute. Therefore, the Cartesian-based approach can not guarantee precise positioning. In summary, for this particular task, none of the three approaches can guarantee precise positioning.

6. CONCLUSION

In this paper, we have defined three different approaches to task re-encoding in vision-based control systems and have shown that these three approaches lead to surprisingly different results on commonly arising problems. These results only scratch the surface of what promises to be an exciting research question—namely, how should one re-encode a task using sensor information so as to guarantee accomplishment of the original task.

The basic results presented here are being extended in a number of ways. Two obvious directions are to extend the analysis to full six degree-of-freedom motion [21], and to address the question of closed-loop performance. In particular, the latter may provide means of comparing the modified Cartesian-based and image-based approaches to determine where they differ. More general tracking problems are also of interest—for example, determining if the problem of exactly matching position *and* velocity of a moving object is a re-encodable task.

Another question which is being studied is the characterization of the set of tasks that can be re-encoded using sensor information so as to guarantee accomplishment of the original task. Invariance, seems to be a sufficient condition {c.f. Lemma 3} but is surely not necessary.

It is likely that a better understanding of these issues will form the basis of a class of adaptive algorithms which learn the calibration of the system online. Knowing the set of tasks which can be re-encoded is central to building a reliable adaptive servoing system.

[5] This is an over-determined system equations; i.e., 3 equations and 2 unknowns. But the assumptions on $\mathcal{F}$ guarantee that the solution (α_1, α_2) always exists.

7. REFERENCES

[1] Seth A. Hutchinson, Gregory D. Hager, and Peter I. Corke. A tutorial on visual servo control. *IEEE Trans. Robot. Automat.*, 12(5):651–670, Oct. 1996.

[2] S. W. Wijesoma, D. F. H. Wolfe, and R. J. Richards. Eye-to-hand coordination for vision-guided robot control applications. *Int. J. Robot. Res.*, 12(1):65–78, Feb. 1993.

[3] Gregory D. Hager, Wen-Chung Chang, and A. Stephen Morse. Robot hand-eye coordination based on stereo vision. *IEEE Contr. Syst. Mag.*, 15(1):30–39, Feb. 1995.

[4] Gregory D. Hager. Calibration-free visual control using projective invariance. In *Proc. Int. Conf. Comput. Vis.*, pages 1009–1015, 1995.

[5] Rafael Kelly. Robust asymptotically stable visual servoing of planar robots. *IEEE Trans. Robot. Automat.*, 12(5):759–766, Oct. 1996.

[6] Lee E. Weiss, Arthur C. Sanderson, and Charles P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE J. Robot. Automat.*, RA-3(5):404–417, Oct. 1987.

[7] Gregory D. Hager. A modular system for robust hand-eye coordination. *IEEE Trans. Robot. Automat.*, 13(4):582–595, 1997.

[8] Claude Samson, Michael Le Borgne, and Bernard Espiau. *Robot Control: The Task Function Approach.* Number 22 in The Oxford engineering science series. Clarendon Press, Oxford, 1991.

[9] Alfred Anthony Rizzi. *Dexterous Robot Manipulation.* PhD thesis, Yale University, Nov. 1995.

[10] Wen-Chung Chang, João Pedro Hespanha, A. Stephen Morse, and Gregory D. Hager. Task re-encoding in vision-based control systems. Technical Report 9701, Center for Computational Vision and Control, March 1997.

[11] Peter K. Allen, Aleksandar Timcenko, Billibon Yoshimi, and Paul Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. Robot. Automat.*, 9(2):152–165, Apr. 1993.

[12] Nicholas Hollinghurst and Robert Cipolla. Uncalibrated stereo hand-eye coordination. *Image and Vision Computing*, 12(3):187–192, Apr. 1994.

[13] Alfred A. Rizzi and Daniel E. Koditscheck. An active visual estimator for dextrous manipulation. *IEEE Trans. Robot. Automat.*, 12(5):697–713, Oct. 1996.

[14] William J. Wilson, Carol C. Williams, and Grahan S. Beli. Relative end-effector control using cartesian position based visual servoing. *IEEE Trans. Robot. Automat.*, 12(5):684–696, Oct. 1996.

[15] John T. Feddema and Owen Mitchell. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. Robot. Automat.*, 5(5):691–700, Oct. 1989.

[16] Steven B. Skaar, William H. Brockman, and W. S. Jang. Three-Dimensional Camera Space Manipulation. *Int. J. Robot. Res.*, 9(4):22–39, Aug. 1990.

[17] Koichi Hashimoto, T. Kimoto, Takumi Ebine, and Hidenori Kimura. Manipulator control with image-based visual servo. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2267–2272, May 1991.

[18] Nikolaos P. Papanikolopoulos, Pradeep K. Khosla, and Takeo Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. Robot. Automat.*, 9(1):14–35, Feb. 1993.

[19] Andrés Castaño and Seth A. Hutchinson. Visual compliance: Task-directed visual servo control. *IEEE Trans. Robot. Automat.*, 10(3):334–342, June 1994.

[20] Horn. *Robot Vision.* The MIT Electrical Engineering and Computer Science Series. McGraw-Hill, New York, 8 edition, 1986.

[21] Wen-Chung Chang. *Vision-Based Control of Uncertain Systems.* PhD thesis, Yale University, Dec. 1997.