What Tasks Can Be Performed with an Uncalibrated Stereo Vision System?*

J. P. Hespanha,† Z. Dodds, G. D. Hager, and A. S. Morse
Center for Computational Vision and Control
c/o Computer Science Department
P.O. Box 208285
Yale University
New Haven, CT, 06520
E-mail: (gregory.hager, joao.hespanha, zachary.dodds, as.morse)@yale.edu

Abstract

This article studies the following question: "When is it possible to decide, on the basis of images of point features observed by an imprecisely modeled two-camera stereo vision system, whether or not a prescribed robot positioning task has been precisely accomplished?" Results are shown for three camera model classes: injective cameras, weakly calibrated projective cameras, and uncalibrated projective cameras. In particular, given a weakly calibrated stereo pair, it is shown that a positioning task can be precisely accomplished if and only if the task specification is invariant to projective transformations. It is shown that injective and uncalibrated projective cameras can accomplish fewer tasks, but are still able to accomplish tasks involving point coincidences.

The same formal framework is applied to the problem of determining the set of tasks which can be precisely accomplished with the well-known position-based control architecture. It is shown that, for any class of camera models, the set of tasks which can be precisely accomplished using a position-based control architecture is a subset of the complete set of tasks which can be decided on the set, but includes all point tasks based on point coincidences. Two ways of extending the idea of position-based control to accomplish more tasks are also presented.

---

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

†João P. Hespanha is now with the Dept. Electrical Eng. & Comp. Science, University of California at Berkeley, 275M Cory Hall #1770, Berkeley, CA 94720-1770. His e-mail is hespanha@robotics.eecs.berkeley.edu.

1 Introduction

Guiding a robotic system using information acquired from cameras has been a central theme of vision and robotics research for decades. In particular, feedback control systems employing video cameras as sensors have been studied in the robotics community for many years (cf. a recent tutorial on visual servoing [25], a review [8] of past work, and a recent workshop [28]). Demonstrated applications of vision within a feedback loop—often referred to as visual servoing or, more generally, *vision-based control*—include automated driving [9], flexible manufacturing [6, 39], and teleoperation with large time delays [11] to name a few.

In order to perform any vision-based control task, two prerequisites must be satisfied. First, it must be possible to *determine* when the task has been accomplished using what is known—namely camera measurements and models of the camera and actuator. Practically speaking, this amounts to being able to define a measurable error signal which, if zero, indicates the task has been accomplished. Second, there must be a control algorithm, driven by this error signal, whose purpose is to move the robot so as to achieve a configuration where the error signal is zero, thus accomplishing the desired the task.

An especially interesting feature of vision-based control systems is that the first requirement can often be satisfied even when camera and actuator models are themselves imprecisely known[1]. As a result, it is possible to position a robot with high accuracy without relying on an accurate hand-eye calibration. This property, which has been observed in several special cases [44, 23, 16, 14, 27, 25, 15, 26, 41], follows from the fact that often *both* the position and orientation of the robot in its workspace *and* the features defining a set of desired poses can be simultaneously observed through the *same* camera system. This observation has motivated a great deal of research on solving the algorithm design problem for imprecisely calibrated or uncalibrated camera-actuator systems [26, 4, 45, 23, 24], and related research on choosing feature configurations to maximize accuracy in the presence of noise [32, 40]. However, little is known about general conditions under which the accomplishment of a given task can be determined in the presence of camera model uncertainty, even though this is a necessary attribute of any vision-based control scheme.

Supposing that the first prerequisite is fulfilled, the form of the control error signal must then be chosen. In a traditional set-point control system, what to choose for a control error is usually clear. In contrast, in vision-based systems there are many choices for a control error, each with different attributes. In particular, it is possible to demonstrate that two control systems using different control errors, both able to perform of the *same* positioning task when the camera system is accurately modeled, can lead to *different* results when the camera system is poorly modeled, even when the camera measurements are themselves error-free. A simple example is the problem of moving a point on a robot manipulator to a position collinear with two reference points in the environment. It is known that, in the absence of measurement noise, this task can be performed with absolute precision with a miscalibrated

[1] This result is in many ways analogous to a conventional set-point control system with a loop-integrator and fixed exogenous reference.

hand-eye system. Yet, it is not hard to exhibit very natural and intuitive control algorithms which could accomplish this task on a perfectly calibrated system, but cannot accomplish it on a miscalibrated one. Although some of the observations just made are implicit in work extending back more than 10 years [43] as well as more recent work [15, 37], to date there has been little formal study of the attributes of a vision-based control system capable of performing precise positioning in the presence of calibration error.

The aim of this paper is to address the questions raised above. Specifically, our goal is to determine conditions under which it is possible to decide, based solely on the observation of point features by an imprecisely modeled two-camera vision system, whether or not a prescribed positioning task has been accomplished. By a positioning task is meant, roughly speaking, the objective of bringing the pose of a robot to a target in the robot's workspace, where both the robot pose and the target are described in terms of observable features or constructions thereof. Positioning tasks are formally defined as equations of the form $T(f) = 0$ where f is a list of observable point features in the workspace of a robot [37]. We consider a task to be accomplished when the physical configuration of the robot in its environment is exactly consistent with the given task description.

In order to address this question, we introduce the formal notions of *task encoding* and *task decidability*. The concept of a task encoding is discussed briefly in [34], which contains further references. An encoded task is simply an equation of the form $E(y) = 0$ where y is a list of observed point features and E is a function which does not depend on knowledge of the calibration of the underlying two-camera system. A given task is "decidable" if there is an encoding of the task which can be used (in a sense which will be made precise later) to "verify" that the task has been accomplished.

We consider task decidability with respect to three classes of two-camera models. It is first shown in §2.4 that, for two-camera models which are injective (but nothing more), precise conditions can be stated under which a given task is decidable and that there is a "non-trivial" class of tasks which can be accomplished by such systems. We then consider two-camera systems modeled using projective geometry. For such systems it is well-known [12] that, in the absence of measurement noise, by calibrating just using images of point features, it is possible to exactly reconstruct the positions of point features "up to a projective transformation" on the three-dimensional projective space. These findings clearly suggest that, for a so-called "weakly calibrated" two-camera model class, there ought to be a close relationship between the decidability of a given task and the invariant properties of the associated task function under projective transformations. Theorem 1 of §3.3 shows that a given task is decidable on a weakly calibrated two-camera class *if and only if* the task is projectively invariant. Finally, we consider two-camera models where weak calibration is not known and show that, in this case, Theorem 1 also provides necessary conditions for a task to be accomplished.

The immediate implication of this result is that a system designer, faced with a particular problem to solve, can now directly determine whether in fact the problem can be solved without demanding precise calibration. Moreover, these results provide a basis for evaluating the effect of miscalibration on different control architectures used for performing tasks. In particular, in the robotics literature two approaches to vision-based control are common: *position-based* control which computes feedback from estimates (computed from measured data) of geometric quantities (pose, feature position, etc.) in the Cartesian space of the robot and *image-based* control which computes feedback directly from measured image data. In this article, these two approaches and a third more recently introduced idea—the modified Cartesian-based approach proposed in [3]—are discussed from the point of view of task encodings. As a result, we are able to state conditions under which these architectures are able to achieve precise positioning with poorly modeled camera systems.

The remainder of this article is structured as follows. The next section sets out basic nomenclature and definitions needed to formally address the task decidability problem and derives conditions for a task to be decidable for camera systems which are injective, but nothing more. In §3 two-camera systems modeled using projective geometry are discussed and the decidability results are specialized to this case. In §4 the implication of these results for two well-known visual servoing architectures and a recently proposed architecture (§4.2) are discussed. The final section recaps these results and outlines some future research directions.

Notation: Throughout this paper, prime denotes matrix transpose, $\mathbb{R}^m$ is the real linear space of m-dimensional column vectors, and $\mathbb{P}^m$ is the real projective space of one-dimensional subspaces of $\mathbb{R}^{m+1}$. Recall that the elements of $\mathbb{P}^m$ are called *points*, *lines* in $\mathbb{P}^m$ are two-dimensional subspaces of $\mathbb{R}^{m+1}$, and, for $m > 2$, planes are three-dimensional subspaces of $\mathbb{R}^{m+1}$. A point $p \in \mathbb{P}^m$ is said to be *on* a line ℓ (respectively plane ψ) in $\mathbb{P}^m$ if p is a linear subspace of ℓ (respectively ψ) in $\mathbb{R}^{m+1}$. For each nonzero vector $x \in \mathbb{R}^m$, $\mathbb{R}x$ denotes both the one-dimensional linear span of x, and also the point in $\mathbb{P}^m$ which x *represents*. The line in $\mathbb{P}^m$ on which two distinct points $p_1, p_2 \in \mathbb{P}^m$ lie, is denoted by $p_1 \oplus p_2$. The *kernel* of a linear or nonlinear function T, with codomain $\mathbb{R}$, is defined to be the set of points x in T's domain such that $T(x) = 0$. As usual, the *image* of a function $H : \mathcal{Z} \to \mathcal{W}$, written $\operatorname{Im} H$, is the set of points $\{H(z) : z \in \mathcal{Z}\}$. The special Euclidean group of rigid body transformations is denoted by SE(3). We use $\subset$ to denote the non-strict subset relation.

2 Task Decidability

This paper is concerned with the problem of achieving precise control of the pose of a robot which moves in a prescribed *workspace* $\mathcal{W} \subset$ SE(3) using two cameras functioning as a position measuring system. The data available to control the robot consists of the projections, onto the cameras' image planes, of point features[2] on the robot as well as point

[2]Throughout this article, the term "feature" always refers to sets (points, lines, etc.) that are observed by cameras. The observations themselves are referred to as "measured data." A "point feature" may be represented by either a point in $\mathbb{R}^3$ or a point in $\mathbb{P}^3$ (i.e., a one-dimensional subspace of $\mathbb{R}^4$). In the examples

features in the environment. All such features lie within the two-cameras' *joint field of view* $\mathcal{V}$. Typically $\mathcal{V}$ will be taken to be either a nonempty subset of $\mathbb{R}^3$ or of $\mathbb{P}^3$.

For the purposes of this article, "precise control" is taken to mean that it is possible for a control system to determine, using only measured data (and in particular *without* precise knowledge of the underlying two-camera configuration), and in the absence of noise, that the physical configuration of the robot system conforms exactly to the specified pose. In the remainder of this section, we formalize the notion of positioning tasks and proceed to develop results which state general conditions under which it is possible to achieve precise control for very general classes of two-camera systems.

2.1 Tasks

By a positioning task or simply a "task" is meant, roughly speaking, the objective of bringing the pose of a robot to a "target" (a set of desired poses) in $\mathcal{W}$. The target set is in turn specified by defining a set of desired geometric constraints on a list of simultaneously observed point features in $\mathcal{V}$. Methods of defining tasks are implicit in the work of several authors including ourselves [2, 6, 34, 37, 15]. Figure 1 depicts examples of how several robotic positioning tasks can be expressed in terms of such feature constraints.

As in [25, 15, 37], tasks are formally represented as equations to be satisfied. In this article, the term "task function" refers, loosely speaking, to a function which maps ordered sets (i.e., lists) of n simultaneously appearing point features $\{f_1, f_2, \ldots, f_n\}$ in $\mathcal{V}$ into the integer[3] set $\{0, 1\}$. We use an un-subscripted symbol such as f to denote each such list and we henceforth refer to f as a *feature*. In some cases only certain features or lists of point features are of interest (e.g., for $n = 3$, one might want to consider only those lists whose three point features are collinear). The set of all such lists is denoted by $\mathcal{F}$ and is a nonempty subset of the set $\mathcal{V}^n$, the Cartesian product of the joint field of view $\mathcal{V}$ with itself n times. In the sequel we call $\mathcal{F}$ the *admissible feature space*. A *task function* is then a given function T from $\mathcal{F}$ to $\{0, 1\}$. The *task* specified by T is the equation

$$T(f) = 0. \tag{1}$$

In case (1) holds we say that the task is *accomplished at f*. Examples of tasks defined in this manner can be found in [37, 10, 7, 6, 25, 15, 36].

Some examples of task functions which are used later in this article are described below.

which follow, points in $\mathbb{R}^m$ are related to points in $\mathbb{P}^m$ by the injective function $x \longmapsto \mathbb{R}\bar{x}$ where $\bar{x}$ is the vector $\begin{bmatrix} x' & 1 \end{bmatrix}'$ in $\mathbb{R}^{m+1}$. With this correspondence, geometrically significant points in $\mathbb{R}^3$ such as a camera's optical center can be unambiguously represented as points in $\mathbb{P}^3$.

[3]In the literature a "task function" is usually defined to be a mapping to the real line or real vector space with additional properties needed to implement a control system (e.g., continuity, differentiability) using it as an error signal [37]. This corresponds most closely to our *encoded task function* introduced in §2.3. As shown later, a task as defined here, when it can be accomplished using measured data, will have an encoded task function which can, in most cases of interest, be "converted" into a task function as defined by [37].

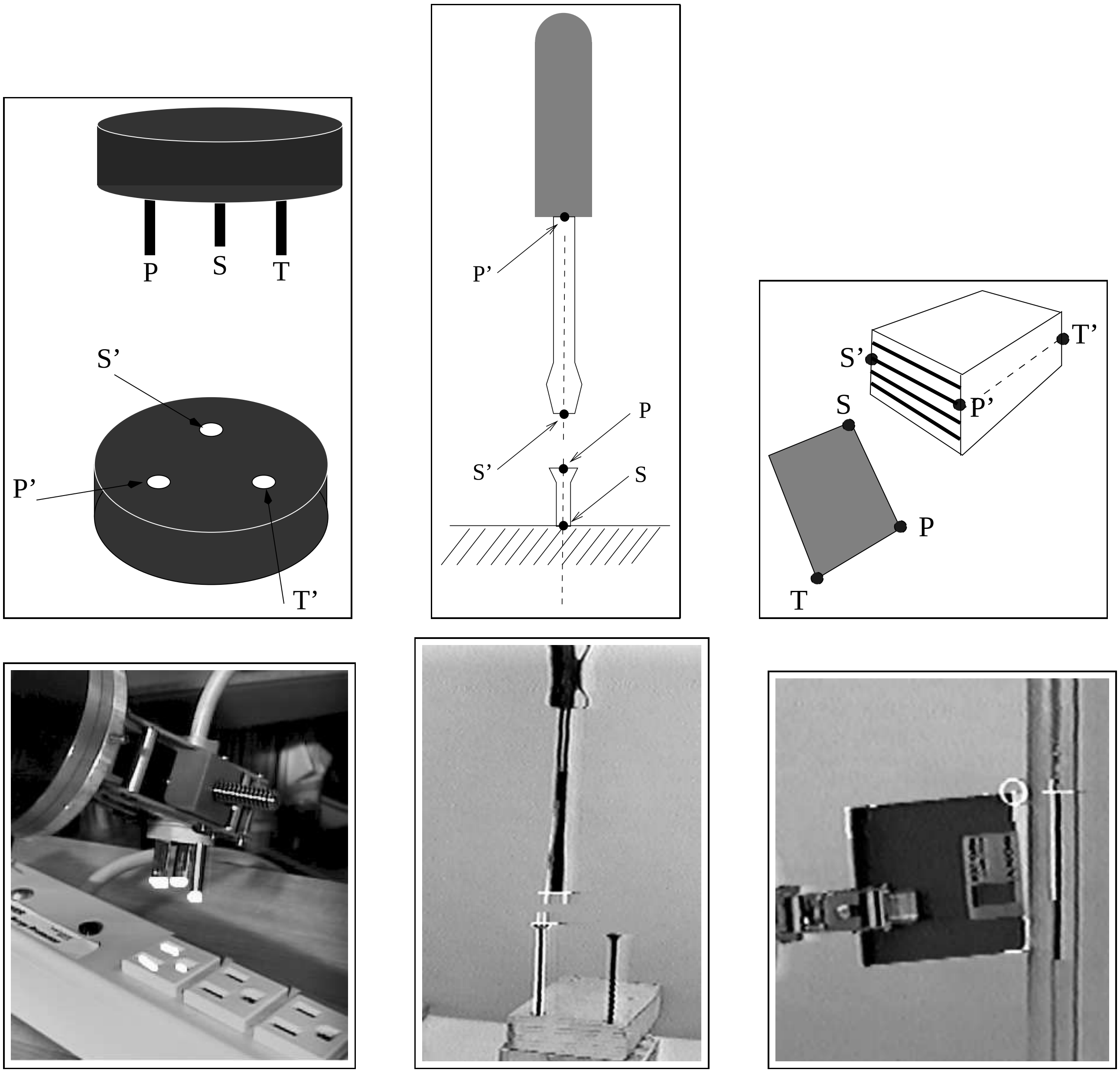

Figure 1: On the top, three schematic examples of tasks specified using point geometry. On left, connecting an electrical connector is specified by making the points P, S, and T coincident with the points P', S' and T' (a conjunction of 3 "point-coincidence tasks"). On the right, a screwdriver is aligned with a screw by making P, S, P', and S' collinear (a conjunction of two "collinearity tasks"). On the bottom, a board is moved to a point where it can be inserted into a rack by first making P, S, T, P', and S' be coplanar and than adding the constraint that P' and S' be between P and S. Note this does not imply that the board is parallel to the rack. Such a constraint could also be specified in terms of point geometry, by using the fact that the sides of the enclosure are parallel to the rack yielding the point T'. One the bottom, physical realizations of these three tasks showing the visual information used to perform them (taken from [15]).

We refer the reader to [15, 17, 42] for examples of these tasks implemented within a visual servoing framework.

Let $\mathcal{V}$ be a subset of $\mathbb{P}^3$. T_{pp} designates the *point-to-point task function*, which is defined on $\mathcal{F}_{\mathrm{pp}} \triangleq \mathcal{V} \times \mathcal{V}$ by the rule

$$\{f_1, f_2\} \longmapsto \begin{cases} 0 & f_1 \text{ and } f_2 \text{ are the same point in } \mathbb{P}^3 \\ 1 & \text{otherwise} \end{cases}$$

In the sequel, we refer to the task specified by T_{pp} as the *point-to-point task.*

T_{3pt} is a task function defined on $\mathcal{F}_{\mathrm{3pt}} \triangleq \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ by the rule

$$\{f_1, f_2, f_3\} \longmapsto \begin{cases} 0 & \text{the } f_i\text{'s are all collinear in } \mathbb{P}^3 \\ 1 & \text{otherwise} \end{cases}$$

We often refer to the task specified by T_{3pt} as the *collinearity task.* Finally, T_{cp} is a task function defined on $\mathcal{F}_{\mathrm{cp}} \triangleq \mathcal{V} \times \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ by the rule

$$\{f_1, f_2, f_3, f_4\} \longmapsto \begin{cases} 0 & \text{the } f_i\text{'s are all coplanar in } \mathbb{P}^3 \\ 1 & \text{otherwise} \end{cases}$$

We often refer to the task specified by T_{cp} as the *coplanarity task.* Note that all of the tasks depicted in Figure 1 have been constructed by appropriate conjunctions of these task functions.

2.2 Problem Formulation

Point features are mapped into the two cameras' *joint image space* $\mathcal{Y}$ through a fixed but imprecisely known *two-camera model* $C_{\mathrm{actual}} : \mathcal{V} \to \mathcal{Y}$ where, depending on the problem formulation, $\mathcal{Y}$ may be either $\mathbb{R}^2 \oplus \mathbb{R}^2$ or $\mathbb{P}^2 \times \mathbb{P}^2$. Typically several point features are observed all at once. If f_i is the ith such point feature in $\mathcal{V}$, then f_i's observed position in $\mathcal{Y}$ is given by the measured output vector $y_i = C_{\mathrm{actual}}(f_i)$. We model miscalibration by assuming that the actual two-camera model C_{actual} is a fixed but unknown element of a prescribed set of injective functions $\mathcal{C}$ which map $\mathcal{V}$ into $\mathcal{Y}$. In the sequel $\mathcal{C}$ is called the *set of admissible two-camera models.*

For the present, no constraints are placed on the elements of $\mathcal{C}$ other than they be injective functions mapping $\mathcal{V}$ into $\mathcal{Y}$. Thus, the results of this section apply to a wide variety of camera projection models including, for example, projection models with lens distortions, spherical projection, and various types of panoramic cameras.

In order to complete the problem formulation, it is helpful to introduce the following nomenclature. For each C in $\mathcal{C}$, let $\bar{C}$ denote the function from $\mathcal{F} \subset \mathcal{V}^n$ to the set $\mathcal{Y}^n$, the

Cartesian product of joint image space $\mathcal{Y}$ with itself n times, which is defined by the rule

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

We call $\bar{C}$ the *extension* of C to $\mathcal{F}$. The aim of this paper is then to give conditions which enable one to decide on the basis of the a priori information, namely $\mathcal{C}$, T, $\mathcal{V}$, and the measured data

$$y \triangleq \bar{C}_{\text{actual}}(f) \tag{2}$$

whether or not the task (1) has been accomplished.

2.3 Task Encodings

Clearly, if C_{actual} were precisely known, a strategy for determining whether or not a task as defined by (1) had been accomplished would be to evaluate (1) using feature locations computed from observed data. However, C_{actual} is not known precisely and, as a result, there are many positioning tasks whose accomplishment cannot be determined from measured data without reliance on an accurate camera calibration, e.g., tasks involving metrical quantities or direct comparisons of Euclidean distances. On the other hand, it is not hard to show that the accomplishment of tasks which involve only point-coincidences can be determined with absolute precision in the presence of camera calibration error (cf. Section 2.4).

It is precisely this dichotomy which motivates us to characterize the nature of tasks whose accomplishment can be determined with imprecisely calibrated two-camera systems. However, in order to make this question precise, we need to first formalize the notion of making decisions as to whether a task has been accomplished using only measured data. Toward this end, let us call a function $E : \mathcal{Y}^n \to \mathbb{R}$ an *encoded task function.* With E so constructed, the equation

$$E(y) = 0 \tag{3}$$

is said to be an *encoded task* or simply an encoding. In case (3) holds we say that the encoded task is *accomplished at* y. In an image-based feedback control system, $E(y)$ would thus be a logical choice for a positioning error [25].

In general, the goal is to choose an encoding so that the accomplishment of the task under consideration at a particular feature configuration is equivalent to accomplishment of the encoded task at the corresponding two-camera measurement. Formally, we say that a task $T(f) = 0$ is *verifiable on* $\mathcal{C}$ with an encoding $E_T(y) = 0$ if,

$$T(f) = 0 \qquad \Longleftrightarrow \qquad E_T(y)|_{y=\bar{C}(f)} = 0, \quad \forall\, f \in \mathcal{F},\ \forall\, C \in \mathcal{C} \tag{4}$$

In other words, $T(f) = 0$ is verifiable on $\mathcal{C}$ with a given encoding $E_T(y) = 0$, if for each feature $f \in \mathcal{F}$ and each admissible two-camera model C in $\mathcal{C}$, the task $T(f) = 0$ is accomplished at

f just in case the encoded task $E_T(y) = 0$ is accomplished at $y = \bar{C}(f)$. In short, in order to verify a given task, an encoding must be able to determine, *even in the face of two-camera model uncertainty,* whether or not the encoded task has been accomplished. Moreover, the encoding must have this property at every point in the joint field of view.

Another characterization of verifiability can be obtained by noting that

$$E_T(y)|_{y=\bar{C}(f)} = (E_T \circ \bar{C})(f)$$

where $E_T \circ \bar{C}$ is the composition of E_T with $\bar{C}$. From this and the definition of verifiability, it follows that $T(f) = 0$ is verifiable on $\mathcal{C}$ with $E_T(y) = 0$ just in the case that for each $C \in \mathcal{C}$, the set of features which T maps into zero is the same as the set of features which $E_T \circ \bar{C}$ maps into zero. We can thus state the following.

Lemma 1 (Verifiability) *Let T, $\mathcal{C}$ and E_T be fixed. Then $T(f) = 0$ is verifiable on $\mathcal{C}$ with the encoded task $E_T(y) = 0$ if and only if*

$$\operatorname{Ker} T = \operatorname{Ker} E_T \circ \bar{C}, \quad \forall C \in \mathcal{C} \tag{5}$$

For example, the task $T_{\text{pp}}(f) = 0$ can be verified on any set of admissible two-camera models by the encoding $E_{\text{pp}}(y) = 0$ that is specified by the encoded task function E_{pp} defined on $\mathcal{Y} \times \mathcal{Y}$, with $\mathcal{Y} = \mathbb{P}^2 \times \mathbb{P}^2$, by the rule

$$\{y_1, y_2\} \longmapsto \begin{cases} 0 & y_1 \text{ and } y_2 \text{ are the same point in } \mathbb{P}^2 \times \mathbb{P}^2 \\ 1 & \text{otherwise} \end{cases}$$

At the end of the next section, we define a much richer class of "point-coincidence" tasks and describe the structure of encodings which verify them.

2.4 Task Decidability

With the preceding definitions in place, our original question can be stated more precisely: "When, for a given set of two-camera models $\mathcal{C}$, can a task be verified?" Here, $\mathcal{C}$ represents the extent of the uncertainty about the actual two-camera model.

To address this question, let us call a given task T *decidable on $\mathcal{C}$* if there exists an encoding $E_T(y) = 0$ for which (4) holds. Thus, the notion of decidability singles out precisely those tasks whose accomplishment (or lack thereof) can be deduced from measured data without regard to a particular encoding which they might be verified with.

The following technical result, which is proven in the appendix, is useful in further characterizing decidable tasks.

Lemma 2 (Decidability) *A given task $T(f) = 0$ is decidable on $\mathcal{C}$ if and only if for each pair $C_1, C_2 \in \mathcal{C}$ and for each pair $f, g \in \mathcal{F}$*

$$\bar{C}_1(f) = \bar{C}_2(g) \Rightarrow T(f) = T(g). \tag{6}$$

That is, a given task is decidable on $\mathcal{C}$ if, when we fix two two-camera models in $\mathcal{C}$, T is constant on any pair of features where the measurements of the two two-camera models agree.

Lemma 2 thus ties the notion of decidability to the properties of the task and the class of admissible two-camera models upon which an encoding of the task would be constructed, *without requiring the construction of such an encoding.*

Point-Coincidence Tasks

As an example of how this lemma may be used, we show here that the family of "point-coincidence tasks", defined below, is decidable (and therefore can be performed with absolute precision) on *any* family of injective two-camera models. To specify what we mean by the family of "point-coincidence tasks" we need to define a few task construction primitives[4]. Given a task $T(f) = 0$, we call the task $\neg T(f) = 0$ specified by

$$\neg T(f) \triangleq 1 - T(f),$$

the *complement* of $T(f) = 0$. Given a permutation $\pi \triangleq \{\pi(1), \pi(2), \ldots, \pi(n)\}$ of the set $\{1, 2, \ldots, n\}$, we call the task $\pi T(f) = 0$ specified by

$$\pi T(f) \triangleq T\big(\pi f\big)$$

with $\pi f \triangleq \{f_{\pi(1)}, f_{\pi(2)}, \ldots, f_{\pi(n)}\}$, the *$\pi$-permutation* of $T(f) = 0$. Also, given two tasks $T_1(f) = 0$ and $T_2(f) = 0$, we call the task $(T_1 \vee T_2)(f) = 0$ specified by

$$(T_1 \vee T_2)(f) \triangleq T_1(f)T_2(f)$$

the *disjunction* of $T_1(f) = 0$ and $T_2(f) = 0$. The following result is straightforward to verify.

Proposition 1 *Given any set of admissible two-camera models $\mathcal{C}$, the following statements are true:*

1. *The complement of a task decidable on $\mathcal{C}$ is decidable on $\mathcal{C}$.*
2. *For any permutation π of $\{1, 2, \ldots, n\}$, the π-permutation of a task decidable on $\mathcal{C}$ is decidable on $\mathcal{C}$.*
3. *The disjunction of two tasks that are decidable on $\mathcal{C}$ is decidable on $\mathcal{C}$.*

[4]These and other task construction primitives are of interest on their own and they will be the topic of a future paper. See also [21].

The family of *point-coincidence tasks on* $\mathcal{F} \subset \mathcal{V}^n$ can then be defined as the smallest set of tasks that contains the task $T_{\text{pc}}(f) = 0$, specified by

$$\{f_1, f_2, \ldots, f_n\} \longmapsto \begin{cases} 0 & f_1 = f_2 \\ 1 & f_1 \neq f_2 \end{cases} \tag{7}$$

and that is closed under task complement, permutation, and disjunction. In short, the family of point-coincidence tasks on $\mathcal{V}^n$ contains any task that can be fully specified by point-coincidence relationships on the n feature points. For example, on $\mathcal{V}^3$,

$$T(f_1, f_2, f_3) = T_{\text{pp}}(f_1, f_2)\, T_{\text{pp}}(f_1, f_3)$$

specifies the task that is accomplished when f_1 is coincident with *at least one of* f_2 or f_3.

Let now $\mathcal{C}$ be an arbitrary set of admissible two-camera models and take a pair of features $f, g \in \mathcal{F}$ and a pair of two-camera models $C_1, C_2 \in \mathcal{C}$ such that

$$\bar{C}_1(f) = \bar{C}_2(g) \tag{8}$$

Suppose first that $T_{\text{pc}}(f) = 0$ and therefore that $f_1 = f_2$. In this case (8) implies that $C_2(g_1) = C_1(f_1) = C_1(f_2) = C_2(g_2)$. This and the injectivity of C_2 guarantee that $g_1 = g_2$ and therefore that $T_{\text{pc}}(g) = 0$. Similarly one can conclude that $T_{\text{pc}}(g) = 1$ whenever $T_{\text{pc}}(f) = 1$. Thus, by Lemma 2, the task $T_{\text{pc}}(f) = 0$ is decidable on $\mathcal{C}$. The following lemma follows from this and Proposition 1.

Lemma 3 *Any point-coincidence task is decidable on any family of admissible two-camera models.*

Although we can prove the decidability of the class of point-coincidence tasks without recourse to encodings, the constructive definition of the class of point-coincidence tasks makes it simple to define encodings for every member of the class. We first define $E_{\text{pc}}(y)$ on $\mathcal{Y}^n$ by the rule

$$\{y_1, y_2, \ldots, y_n\} \longmapsto \begin{cases} 0 & y_1 = y_2 \\ 1 & y_1 \neq y_2 \end{cases}. \tag{9}$$

It is simple to verify that $E_{\text{pc}}(y) = 0$ verifies $T_{\text{pc}}(f) = 0$ on any family $\mathcal{C}$ of admissible two-camera models. Furthermore, if we define complement, permutation, and disjunction of encodings in the same manner we defined these operations for tasks, then one can obtain an encoding that verifies a given point-coincidence task $T(f) = 0$ on $\mathcal{C}$ by applying the same construction to the encoding that was used to build the task $T(f) = 0$ from the task $T_{\text{pc}}(f) = 0$ (starting now with the encoded task function $E_{\text{pc}}(y) = 0$). This gives a systematic procedure to build encodings that verify any point-coincidence task on any family of admissible two-camera models.

3 Task Decidability for Projective Camera Models

The results of the preceding section apply to two-camera model classes whose elements are assumed to be injective but nothing more. In this section we specialize to the case when the two-camera models of interest are pairs of projective camera models which map subsets of $\mathbb{P}^3$ containing $\mathcal{V}$, into $\mathbb{P}^2 \times \mathbb{P}^2$. Projective models of this type have been widely used in computer vision [31, 13, 18, 19, 30, 1] in part because they include as special cases the perspective, affine, and orthographic camera models. By restricting attention to projective models, we are able to provide a complete and concise characterization of decidable tasks in terms of their invariance under the group of projective transformations [31].

3.1 Uncalibrated Stereo Vision Systems

In order to formalize the notion of a camera as a sensor, we need to delineate both the structure of the camera mapping *and* the domain over which it operates. Although the general structure of a projective camera model is well known in computer vision [31, 13], it is important to realize that it is not sufficient, for this analysis, to consider cameras simply as maps from $\mathbb{P}^3$ to $\mathbb{P}^2$. To do so would introduce models with singular points (points at which injectivity breaks down) as well as points in $\mathbb{P}^3$ which have no rendering in terms of the underlying physical system that we are modeling [13].

Thus, we proceed as follows. For any real 3×4 full-rank matrix M, let $\mathbb{P}^3_M$ denote the set of all points in $\mathbb{P}^3$ except for $\operatorname{Ker} M$. We call the function $\mathbf{M}$ from $\mathbb{P}^3_M$ to $\mathbb{P}^2$ defined by the rule $\mathbb{R}x \longmapsto \mathbb{R}Mx$ the *global camera model induced by* M and we call $\operatorname{Ker} M$ the *optical center* of $\mathbf{M}$ [5].

Consider now a pair of global camera models, $\mathbf{M}$ and $\mathbf{N}$, whose optical centers are distinct. We define the *baseline* of the two-camera system to be the line in $\mathbb{P}^3$ on which the optical centers of $\mathbf{M}$ and $\mathbf{N}$ lie, namely $\ell \stackrel{\Delta}{=} \operatorname{Ker} M \oplus \operatorname{Ker} N$. Let $\mathbb{P}^3_\ell$ denote the set of all points in

[5] When it is possible to write M in the form

$$M = \begin{bmatrix} H & -Hc \end{bmatrix}$$

where H is a nonsingular 3×3 matrix, and c is a vector in $\mathbb{R}^3$, then $\mathbf{M}$ models a projective camera with center of projection at c [13]. In this case the kernel of M is $\mathbb{R}\left[\begin{smallmatrix} c \\ 1 \end{smallmatrix}\right]$ which justifies calling it the optical center of $\mathbf{M}$. One special case occurs when $H = R$, where R is a 3×3 rotation matrix. In this case, $\mathbf{M}$ models a perspective camera with unit focal length, optical center at $c \in \mathbb{R}^3$, and orientation defined by R. On the other hand, it is also possible for M to be of the form

$$M = \begin{bmatrix} \bar{R} & -\bar{R}c \\ 0_{1\times 3} & 1 \end{bmatrix}$$

where $\bar{R} \stackrel{\Delta}{=} \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right] R$ and R is a rotation matrix. In this case, $\mathbf{M}$ models an orthographic camera whose coordinate frame is defined by the rotation matrix R. The kernel of M is $\mathbb{R}\left[\begin{smallmatrix} r_3 \\ 0 \end{smallmatrix}\right]$ where r_3 is the third column of R'. This model also can be regarded as a perspective camera model with optical center at infinity and "orientation" defined by the z-axis of the coordinate frame.

$\mathbb{P}^3$ except for those that lie on ℓ. We call the function $G = \{\mathbf{M}, \mathbf{N}\}$ from $\mathbb{P}^3_\ell$ to the image space $\mathcal{Y} \triangleq \mathbb{P}^2 \times \mathbb{P}^2$, defined by the rule $\mathbb{R}x \longmapsto \{\mathbb{R}Mx, \mathbb{R}Nx\}$, the *global two-camera model*[6] *induced by* $\{M, N\}$.

As a consequence of these definitions, it is straightforward to show that any global two-camera model so defined is injective on $\mathbb{P}^3_\ell$. However, as suggested above, the domain $\mathbb{P}^3_\ell$ is still larger than we might desire. For example, if we took the field of view of the cameras to be $\mathbb{P}^3_\ell$ for some line ℓ in $\mathbb{P}^3$, then the set of two-camera models consistent with this field of view would be reduced to only those two-camera models whose baseline is precisely ℓ. In fact, it will turn out later (specifically Lemma 5 of the next section) that we need to exclude from $\mathcal{V}$ at least the points on one plane in $\mathbb{P}^3$. Although the choice of this plane (or any larger set for that matter) is arbitrary, we will specifically restrict our attention to two-camera models whose fields of view are a given subset $\mathcal{V} \subset \mathbb{P}^3$ of the form

$$\mathcal{V} \triangleq \left\{ \mathbb{R} \begin{bmatrix} w \\ 1 \end{bmatrix} : w \in \mathcal{B} \right\}$$

where $\mathcal{B}$ is a nonempty subset of $\mathbb{R}^3$. Under the canonical mapping between $\mathbb{R}^4$ and $\mathbb{P}^3$, we exclude by this construction the points in the "plane at infinity", namely points in $\mathbb{P}^3$ which have no corresponding point in $\mathbb{R}^3$.

In the sequel we say that the baseline of a global two-camera model G lies outside of $\mathcal{V}$ if there is no point on the baseline of G which is also in $\mathcal{V}$. For each such G it is possible to define a restricted function from $\mathcal{V}$ to $\mathcal{Y}$ by the rule $v \longmapsto G(v)$. We denote this function by $G|\mathcal{V}$ and refer to it as the *two-camera model determined by* G *on* $\mathcal{V}$. By the set of all *uncalibrated two-camera models on* $\mathcal{V}$, written $\mathcal{C}_{\text{uncal}}[\mathcal{V}]$, is meant the set of all two-camera models which are determined by global two-camera models whose baselines lie outside of $\mathcal{V}$. For the remainder of this article, any stereo vision system whose cameras admit a model which is known to be in this class but is otherwise unknown, is said to be *uncalibrated*.

3.2 Weakly Calibrated Stereo Vision Systems

It is well-known that, given measurements of a sufficient number of points in "general position" by a stereo camera system, it is possible to compute a one-dimensional constraint on the (stereo) projection of any point features in the two-camera field of view [29, 13, 33, 46]. A stereo camera system for which this information, the "epipolar constraint," is known is often referred to as *weakly calibrated* [35].

It follows that we can view weak calibration as a restriction on the class of two-camera models which can be characterized as follows. Let us write GL(4) for the general linear group of real, nonsingular, 4×4 matrices. For each such matrix A, $\mathbf{A}$ denotes the corresponding projective transformation defined on $\mathbb{P}^3 \to \mathbb{P}^3$ by the rule $\mathbb{R}x \longmapsto \mathbb{R}Ax$. For each fixed global

[6]Henceforth, the term "two-camera model" is taken to mean a projective two-camera model.

two-camera model $G_0 \triangleq \{\mathbf{M_0}, \mathbf{N_0}\}$ whose baseline ℓ lies outside of $\mathcal{V}$, let $\mathcal{C}[G_0]$ denote the set of two-camera models

$$\mathcal{C}[G_0] \triangleq \{(G_0A)|\mathcal{V} : A \in \mathrm{GL}(4),\ \text{and } \mathbf{A}(\mathcal{V}) \subset \mathbb{P}^3_\ell\}$$

where G_0A is the global two-camera model induced by $\{M_0A, N_0A\}$. Note that for any $A \in \mathrm{GL}(4)$ such that $\mathbf{A}(\mathcal{V}) \subset \mathbb{P}^3_\ell$, $(G_0A)|\mathcal{V}$ is well defined and is equal to $G_0 \circ (\mathbf{A}|\mathcal{V})$ where $\mathbf{A}|\mathcal{V}$ is the restricted function $\mathcal{V} \to \mathbb{P}^3_\ell$, $v \longmapsto \mathbf{A}(v)$. Thus we can also write

$$\mathcal{C}[G_0] = \{G_0 \circ (\mathbf{A}|\mathcal{V}) : A \in \mathrm{GL}(4),\ \text{and } \mathbf{A}(\mathcal{V}) \subset \mathbb{P}^3_\ell\} \tag{10}$$

It is easy to verify that for any G_0 in $\mathcal{C}_{\mathrm{uncal}}[\mathcal{V}]$, $\mathcal{C}[G_0]$ is contained within $\mathcal{C}_{\mathrm{uncal}}[\mathcal{V}]$.

The definition of the set $\mathcal{C}[G_0]$ is designed to include all two-camera models which are indistinguishable from G_0 based on observed data—that is, all those that have the same weak calibration as G_0. Alternatively, $\mathcal{C}[G_0]$ could also have been defined as the set of all global two-camera models that are injective on $\mathcal{V}$ and have the same fundamental matrix [13] as G_0. As is stated in the following lemma, these two definitions are, in fact, equivalent.

Lemma 4 *Let G_0 be a given global two-camera model whose baseline lies outside $\mathcal{V}$. The set $\mathcal{C}[G_0]$ contains all two-camera models that are injective on $\mathcal{V}$ and have the same fundamental matrix as G_0.*

One implication of Lemma 4 is that for a camera pair with known fundamental matrix, the two-camera model is known up to a right-composition with a (suitably restricted) projective transformation. Therefore, reconstruction done with such a system can only be correct up to a projective transformation on the point-features as was reported in [12].

Proof of Lemma 4: First note that a global two-camera model G has the same epipolar geometry as G_0 just in case $\operatorname{Im} G = \operatorname{Im} G_0$. This is because it is the epipolar constraint—and therefore the fundamental matrix—that determines which points in the codomain $\mathcal{Y} \times \mathcal{Y}$ of a global two-camera model are actually in its image. On the other hand, it is well known that given any two functions $G_0 : \mathbb{P}^3_\ell \to \mathcal{Y} \times \mathcal{Y}$ and $G : \mathbb{P}^3_{\ell_1} \to \mathcal{Y} \times \mathcal{Y}$, the fact that $\operatorname{Im} G = \operatorname{Im} G_0$ is equivalent to the existence of an injective function $X : \mathbb{P}^3_{\ell_1} \to \mathbb{P}^3_\ell$ such that $G = G_0 \circ X$. Moreover, if G and G_0 are global two-camera models, their linear structure requires the function X to be of the form $\mathbb{R}x \longmapsto \mathbb{R}Ax$ for some $A \in \mathrm{GL}(4)$.

The two equivalences stated above allow one to conclude that a global two-camera model G has the same epipolar geometry as G_0 just in case there is a matrix $A \in \mathrm{GL}(4)$ such that $G = G_0 \circ (\mathbf{A}|\mathbb{P}^3_{\ell_1})$, where ℓ_1 denotes the baseline of G and $\mathbf{A}|\mathbb{P}^3_{\ell_1}$ denotes the restriction $\mathbb{P}^3_{\ell_1} \to \mathbb{P}^3_\ell$, $v \longmapsto \mathbf{A}(v)$. Lemma 4 then follows from this and the fact that $G_0 \circ (\mathbf{A}|\mathbb{P}^3_{\ell_1})$ is injective on $\mathcal{V}$ just in case $\mathbf{A}(\mathcal{V}) \subset \mathbb{P}^3_\ell$. $\blacksquare$

Finally, the following technical result will be needed:

Lemma 5 *Let G_0 be a given global two-camera model whose baseline lies outside $\mathcal{V}$. For any matrix $A \in \mathrm{GL}(4)$, there always exists a global two-camera model G_1 such that both $G_1|\mathcal{V}$ and $G_1 \circ (\mathbf{A}|\mathcal{V})$ are in $\mathcal{C}[G_0]$.*

The proof of this lemma, which is included in the appendix, depends on the fact that there is at least one plane in $\mathbb{P}^3$ which is not in $\mathcal{V}$—hence our definition of the field of view as excluding the plane at infinity.

3.3 Decidability for Weakly Calibrated Stereo Vision Systems

As noted above, it has been shown that, with a "weakly calibrated" stereo system, it is possible to reconstruct the position of point features in the two-cameras' field of view from image measurements. However, this reconstruction is only unique up to a projective transformation [12]. These findings suggest that there should be a connection between the decidability of a task $T(f) = 0$ on a weakly calibrated two-camera class and the properties of $T(f) = 0$ which are invariant under projective transformations. To demonstrate that this is so, we begin by defining what is meant by a "projectively invariant task". For each $A \in \mathrm{GL}(4)$, let $\bar{\mathbf{A}}$ denote the extended function from $(\mathbb{P}^3)^n$ to $(\mathbb{P}^3)^n$ defined by the rule

$$\{p_1, p_2, \dots, p_n\} \longmapsto \{\mathbf{A}(p_1), \mathbf{A}(p_2), \dots \mathbf{A}(p_n)\}$$

where $(\mathbb{P}^3)^n$ denotes the Cartesian product of $\mathbb{P}^3$ with itself n times. We call two points f and g in $\mathcal{F}$ *projectively equivalent*[7] if there exists an A in GL(4) such that $f = \bar{\mathbf{A}}(g)$. A task $T(f) = 0$ is said to be *projectively invariant on* $\mathcal{F}$ if for each pair of projectively equivalent points $f, g \in \mathcal{F}$,

$$T(f) = T(g) \tag{11}$$

In other words, $T(f) = 0$ is projectively invariant on $\mathcal{F}$, just in case T is constant on each equivalence class of projectively equivalent features within $\mathcal{F}$. It follows that T is a projective invariant [31].

The main result of this section is the following:

Theorem 1 (Weak Calibration) *Let G_0 be a fixed global two-camera model whose baseline lies outside of $\mathcal{V}$. A task $T(f) = 0$ is decidable on $\mathcal{C}[G_0]$ if and only if it is projectively invariant.*

In short, with a weakly calibrated two-camera system, any projectively invariant task is verifiable with at least one encoding. Moreover, any task which is not projectively invariant is **not** verifiable with any type of encoding.

In the proof which follows, for each global two-camera $G : \mathbb{P}^3_\ell \to \mathcal{Y}$, $\bar{G}$ denotes the extended function from $(\mathbb{P}^3_\ell)^n$ to $\mathcal{Y}^n$ which is defined by the rule

$$\{p_1, p_2, \dots, p_n\} \longmapsto \{G(p_1), G(p_2), \dots, G(p_n)\}$$

[7]Note that projective equivalence is an equivalence relation on $\mathcal{F}$ because the set of extensions to $(\mathbb{P}^3)^n$ of all projective transformations on $\mathbb{P}^3$, is a group with composition rule $(\bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2) \longmapsto \bar{\mathbf{A}}_1 \circ \bar{\mathbf{A}}_2$.

Here $(\mathbb{P}^3_\ell)^n$ denotes the Cartesian product of $\mathbb{P}^3_\ell$ with itself n times.

Proof of Theorem 1: Suppose that $T(f) = 0$ is projectively invariant. To prove that $T(f) = 0$ is decidable on $\mathcal{C}[G_0]$ let $f, g \in \mathcal{F}$ be any pair of features such that $\bar{C}_1(f) = \bar{C}_2(g)$, where $\bar{C}_1$ and $\bar{C}_2$ are extensions of some two-camera models $C_1, C_2 \in \mathcal{C}[G_0]$. By Lemma 2, it is enough to show that $T(f) = T(g)$.

Since $C_1, C_2 \in \mathcal{C}[G_0]$, there must be matrices $A_i \in \mathrm{GL}(4)$ such that $C_i = \{G_0 \circ \mathbf{A}_i\}|\mathcal{V}$, $i \in \{1, 2\}$ and

$$\mathbf{A_i}(\mathcal{V}) \subset \mathbb{P}^3_\ell, \quad i \in \{1, 2\} \tag{12}$$

where ℓ is the baseline of G_0. Thus $\bar{C}_1(f) = \bar{G}_0(\bar{\mathbf{A}}_\mathbf{1}(f))$ and $\bar{C}_2(g) = \bar{G}_0(\bar{\mathbf{A}}_\mathbf{2}(g))$. Hence $\bar{G}_0(\bar{\mathbf{A}}_\mathbf{1}(f)) = \bar{G}_0(\bar{\mathbf{A}}_\mathbf{2}(g))$. Now each point feature in the list $\bar{\mathbf{A}}_\mathbf{1}(f)$ and in the list $\bar{\mathbf{A}}_\mathbf{2}(g)$ must be in $\mathbb{P}^3_\ell$ because of (12). Thus $\bar{\mathbf{A}}_\mathbf{1}(f)$ and $\bar{\mathbf{A}}_\mathbf{2}(g)$ must be points in $(\mathbb{P}^3_\ell)^n$. It follows that $\bar{\mathbf{A}}_\mathbf{1}(f) = \bar{\mathbf{A}}_\mathbf{2}(g)$ since G_0 is injective. This means that $f = \{\bar{\mathbf{A}}_\mathbf{1}^{-1} \circ \bar{\mathbf{A}}_\mathbf{2}\}(g)$ and thus that f and g are projectively equivalent. Hence $T(f) = T(g)$.

To prove the converse, now suppose that $T(f) = 0$ is decidable on $\mathcal{C}[G_0]$. Pick two projectively equivalent features $f, g \in \mathcal{F}$ and a matrix $A \in \mathrm{GL}(4)$ such that $f = \bar{\mathbf{A}}(g)$. By Lemma 5 there are two two-camera models in $\mathcal{C}[G_0]$ that can be written as $C_1 = G_0 \circ (\mathbf{B}|\mathcal{V})$ and $C_2 = G_0 \circ (\mathbf{B} \circ \mathbf{A}|\mathcal{V})$, with $B \in \mathrm{GL}(4)$. Then, it follows that $\bar{C}_1(f) = \bar{C}_2(g)$ and therefore that $T(f) = T(g)$, because of Lemma 2 and the hypothesis of decidability on $\mathcal{C}[G_0]$. ■

3.4 Decidability for Uncalibrated Stereo Vision Systems

As it stands, Theorem 1 applies only to stereo vision systems which are weakly calibrated. However, since being projectively invariant is a necessary condition for the task $T(f) = 0$ to be decidable on $\mathcal{C}[G_0]$, projective invariance must also be a necessary condition for the task $T(f) = 0$ to be decidable on $\mathcal{C}_{\mathrm{uncal}}[\mathcal{V}]$. We can thus state the following.

Corollary 1 *If $T(f) = 0$ is decidable on $\mathcal{C}_{\mathrm{uncal}}[\mathcal{V}]$, then $T(f) = 0$ is projectively invariant.*

The reverse implication, namely that a task's projective invariance implies decidability on $\mathcal{C}_{\mathrm{uncal}}$, is false. One example is the collinearity task, $T_{3\mathrm{pt}}(f) = 0$ defined in Section 2.1. Clearly, the task $T_{3\mathrm{pt}}(f) = 0$ is projectively invariant because projective transformations preserve collinearity. Yet, this task is not decidable on $\mathcal{C}_{\mathrm{uncal}}$. Indeed, there is a pair of two-camera models $C_1, C_2 \in \mathcal{C}_{\mathrm{uncal}}$ and pairs of features $f, g \in \mathcal{F}$ at which $\bar{C}_1(f)$ and $\bar{C}_2(g)$ are equal and yet the task is accomplished at f but not at g. This happens when all point features in the list f and the optical centers of the global two-camera models that determine C_1 are contained in a single 3-dimensional subspace of $\mathbb{R}^4$ and also when all point features in the list g and the optical centers of the global two-camera models that determine C_2 are contained in a single 3-dimensional subspace of $\mathbb{R}^4$. In geometric terms, the task fails to be decidable on a given two-camera model when the feature points and the cameras' optical centers all lie in the same epipolar plane. Practically speaking, the task can be accomplished with uncalibrated cameras provided such configurations can be avoided [15].

Another example of a projectively invariant task which is not decidable using uncalibrated two-cameras is coplanarity—that is the task $T_{\mathrm{cp}}(f) = 0$ is not decidable on $\mathcal{C}_{\mathrm{uncal}}$. It is interesting to note that the collinearity task fails to be decidable because there is a "small" set of configurations where the task is not verifiable using any encoding. Thus, it is still possible to verify the accomplishment of this task, without weak calibration, provided the problem can be constrained to avoid such configurations. On the other hand, there is *no* situation (without introducing additional constraints on feature sets) where coplanarity could be decided using uncalibrated two-cameras.

Thus, these results also clarify the value of performing weak calibration. First, weak calibration removes (visually) singular configurations from the robot workspace and in turn makes the problem of designing control algorithms simpler. Second, weak calibration augments the class of decidable tasks.

3.5 Classes of Tasks

One way of viewing our results to this point is as follows. Fix a field of view $\mathcal{V}$ and a feature space $\mathcal{F} \subset \mathcal{V}^n$. Let $\mathcal{T}_{\mathrm{pc}}$ denote the set of all point-coincidence tasks on $\mathcal{F}$ as defined in Section 2.4, let $\mathcal{T}_{\mathrm{PInv}}$ denote the set of all projectively invariant tasks on $\mathcal{F}$, and let $\mathcal{T}_{\mathrm{uncal}}$ and $\mathcal{T}_{\mathrm{wkcal}}$ denote the set of tasks which are decidable on $\mathcal{C}_{\mathrm{uncal}}[\mathcal{V}]$ and $\mathcal{C}[G_0]$ for some $G_0 \in \mathcal{C}_{\mathrm{uncal}}[\mathcal{V}]$, respectively. Then from the preceding analysis, we know that

$$\mathcal{T}_{\mathrm{pc}} \subset \mathcal{T}_{\mathrm{uncal}} \subset \mathcal{T}_{\mathrm{wkcal}} = \mathcal{T}_{\mathrm{PInv}}$$

One might now reasonably ask how rich these sets are. In particular, we were able to *constructively* define the set of point-coincidence tasks, and as a result were also able to construct an encoding for any task in the class. Is this also possible for the sets $\mathcal{T}_{\mathrm{uncal}}$ and $\mathcal{T}_{\mathrm{PInv}}$?

The structure of the set $\mathcal{T}_{\mathrm{PInv}}$ has been studied in [20] with some generality. Here, we take from [21] a characterization for $\mathcal{T}_{\mathrm{PInv}}$ for the special cases when $n \in \{2, 3, 4\}$. For a given set of tasks $\mathcal{T}$, let us say that the task $T(f) = 0$ can be *generated by* $\mathcal{T}$ if it can be obtained by applying any number of the three task construction primitives defined in Section 2.4 to the tasks in $\mathcal{T}$. The following is taken from [21]:

Theorem 2 *Given the feature space $\mathcal{F} = \mathcal{V}^n$, the following statements are true:*

1. *For $n = 2$ any projectively invariant task can be generated by a set consisting of exactly one task, namely the one specified by the point-to-point task function T_{pp}.*

2. *For $n = 3$ any projectively invariant task can be generated by the set of tasks specified by the two task functions $f \longmapsto T_{\mathrm{pp}}(\{f_1, f_2\})$ and T_{3pt}.*

3. *for $n = 4$ any projectively invariant task can be generated by the set of tasks specified by the three task functions $f \longmapsto T_{\mathrm{pp}}(\{f_1, f_2\})$, $f \longmapsto T_{\mathrm{3pt}}(\{f_1, f_2, f_3\})$, and T_{cp}, together with the family of tasks specified by*

$$f \longmapsto \begin{cases} 0 & f_1, f_2, f_3, f_4 \text{ on the same line in } \mathbb{P}^3 \text{ with cross ratio [31] } \rho \\ 1 & \text{otherwise} \end{cases}$$

where ρ takes values in $\mathbb{R}$.

We note that, as with the family of point-coincidence tasks, the constructiveness of these definitions makes it straightforward to construct encodings for every member of each class of tasks.

4 Implications for Vision-Based Control

In the introduction we described two commonly used control system architectures for accomplishing vision-based tasks: position-based control and image-based control. Following the generally accepted definitions developed by [43] and summarized in [25], a position-based control architecture is one in which there is an explicit conversion of measured image features to task-space coordinates of those features. The control problem is then solved in task space. In contrast, an image-based control architecture is one in which there is no explicit conversion of measured image features to workspace coordinates of those features. Instead, a control signal is generated directly from measured image information. It should be noted, however, that even when such an estimate is not used in the encoding, a reasonably good (but not necessarily perfect) estimate of C_{actual} is often necessary in practice. For example, such an estimate may be necessary to construct a feedback controller which provides (at least) loop-stability and consequently a guarantee that the encoded task can be accomplished in practice.

In general, both position-based and image-based control architectures are designed so as to drive to zero a specific "control error", which is a function E of the two-camera measurements y. Thus, E can be regarded as an encoded task function. Therefore, the dichotomy between position-based and image-based architectures is directly related to the form of the encoding used to generate the control error. Specifically, image-based architectures do not commit to any particular structure of the encodings whereas position-based architectures employ encodings that include an explicit conversion of measured image features to workspace coordinates of those features.

Thus, from the results of the previous section, we know that an image-based control architecture can, with the proper choice of encodings, precisely accomplish the widest possible range of tasks. However, what of position-based architectures? In this section, we explore some of the encodings that are used to generate control errors in position-based architectures and point out some of the potential limitations of this approach.

4.1 Cartesian-Based Encoding

The position-based control architecture is, in a sense, motivated by the heuristic idea of "certainty equivalence" as used in control [38, 22]. In the present context, certainty equivalence advocates that to verify a given task $T(f) = 0$, one should evaluate $T(f_{\text{est}}) = 0$ for some estimate f_{est} of f with the understanding that f_{est} is to be viewed as correct even though it may not be.

This idea leads naturally to what we refer to as a *Cartesian-based task encoding.* Again, the idea is to embed an "estimation procedure" in the encoding itself. The construction of such an estimate starts with the selection (usually through an independent calibration procedure) of a two-camera model C_{est} in $\mathcal{C}$ which is considered to be an estimate of C_{actual}. Then, f_{est} is defined by an equation of the form

$$f_{\text{est}} \stackrel{\Delta}{=} \bar{C}_{\text{est}}^{-1}(y)$$

where $\bar{C}_{\text{est}}^{-1}$ is one of the left inverses of $\bar{C}_{\text{est}}$. Note that such a left inverse can always be found because $\bar{C}_{\text{est}}$ is injective. In accordance with certainty equivalence, the *Cartesian-based encoding* of $T(f) = 0$ is then

$$T\big(\bar{C}_{\text{est}}^{-1}(y)\big) = 0 \tag{13}$$

The encoded task function is thus $E_T \stackrel{\Delta}{=} T \circ \bar{C}_{\text{est}}^{-1}$. In light of Lemma 1, it is clear that a given task $T(f) = 0$ will be verifiable on $\mathcal{C}$ by a Cartesian-based encoding of the form

$$(T \circ \bar{C}_{\text{est}}^{-1})(y) = 0$$

just in case

$$\operatorname{Ker} T = \operatorname{Ker} T \circ \bar{C}_{\text{est}}^{-1} \circ \bar{C}, \quad \forall C \in \mathcal{C} \tag{14}$$

It is worth noting that, to encode a given task in this way, it is necessary to pick *both* a model C_{est} from $\mathcal{C}$ and a left inverse of its extension. Because such left inverses are generally not unique, there are in fact many ways to encode $T(f) = 0$ in this manner, even after C_{est} has been chosen.

In practice, the appeal of Cartesian-based encodings is that, in the ideal case of perfect reconstruction, f_{est} is a metrically accurate representation of feature locations in the world. With such a reconstruction, one can phrase any task, even a task relying on an absolute measurement such as "move 3 centimeters along the line joining features f_1 and f_2." Clearly, however, such a task is not projectively invariant. Therefore, based on the results of the previous section, in practice it is not possible to provide guarantees on the accuracy with which such a task would actually be accomplished by using a weakly calibrated two-camera system.

More importantly, as noted above, there are projectively invariant tasks which are not verifiable using an arbitrary Cartesian-based encoding—the collinearity task is one of these. Thus the following statements can be made:

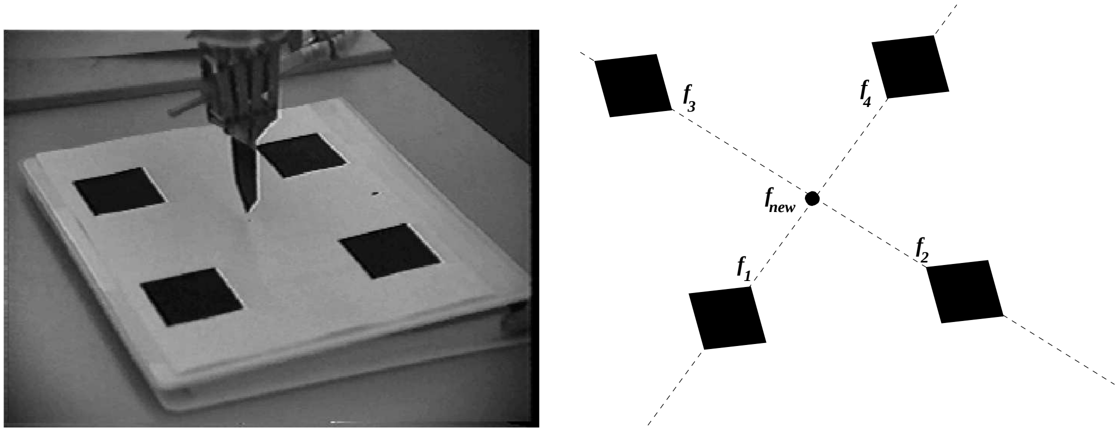

Figure 2: On the left, a midpoint placement task to be performed. On the right, the midpoint construction used to describe the task in a manner which permits accurate task performance.

1. There are projectively invariant tasks which *cannot* be verified on any set of weakly calibrated two-camera systems using an arbitrary Cartesian-based encoding.

2. Any member of the previously defined class of point-coincidence tasks $\mathcal{T}_{\mathrm{pc}}$, can be verified by a Cartesian-based encoding on a given set of admissible two-camera models $\mathcal{C}$, provided that

$$\bar{C}_{\mathrm{est}}^{-1}(y) = \{C_{\mathrm{est}}^{-1}(y_1), C_{\mathrm{est}}^{-1}(y_2), \ldots, C_{\mathrm{est}}^{-1}(y_n)\}, \qquad y \in \mathcal{Y}^n$$

   for some left inverse C_{est}^{-1} of C_{est} for which $C_{\mathrm{est}}^{-1} \circ C$ is injective for every $C \in \mathcal{C}$.

4.2 Modified Cartesian Encoding

Suppose, for a fixed set of projective two-camera models $\mathcal{C}$, we let $\mathcal{T}_{\mathrm{Cbased}}$ and $\mathcal{T}_{\mathrm{Ibased}}$ denote the set of all tasks which can be verified on $\mathcal{C}$ with a Cartesian-based and image-based encodings (as defined above), respectively. The results of this section thus far demonstrate that $\mathcal{T}_{\mathrm{Cbased}}$ is a subset of $\mathcal{T}_{\mathrm{Ibased}}$. A specific example of a task which cannot be performed with an arbitrary Cartesian-based encoding is that of positioning a point at the *midpoint* of four coplanar points as illustrated in Figure 2.

One of the motivations for the use of Cartesian-based encodings is the fact that they use estimates of features (i.e., f_{est}) taking values in Cartesian space which, in turn, is the natural space for specifying robot positioning tasks. One might now reasonably ask if there is a way of "enriching" $\mathcal{T}_{\mathrm{Cbased}}$ so that one might gain the advantage of using more intuitive position-based control while retaining the robustness of image-based control. The "modified" Cartesian-based encoding, introduced in [5], is one way of extending the idea of Cartesian-based encodings to encompass a richer set of tasks.

The central idea in modified Cartesian-based encodings is to reformulate a positioning task, initially defined on some feature set $\mathcal{F} \subset \mathcal{V}^n$, on a new feature space $\mathcal{F}_{\text{new}} \subset \mathcal{V}^m$. In what follows, let $\mathcal{F}_{\text{new}}$ be fixed and write $\bar{C}_{\text{new}}$ for the extension of $C \in \mathcal{C}$ to $\mathcal{F}_{\text{new}}$. We say that a function $H : \mathcal{F} \to \mathcal{F}_{\text{new}}$ *factors through* $\mathcal{C}$ if there is a function K such that

$$\bar{C}_{\text{new}} \circ H = K \circ \bar{C}, \quad \forall C \in \mathcal{C} \tag{15}$$

Intuitively, H is a construction on the feature space which results in "new" features. K is a construction, based on observed data, which results in a "new" measurement in $\mathcal{F}_{\text{new}}$. For example, Figure 2(right) illustrates the geometric construction of the midpoint of four coplanar points. In this case, H_{midpt} would be the function which maps lists of four features (f_1 through f_4 in the figure) into a single feature—the midpoint (f_{new} in the figure). K_{midpt} would be the function which does the same construction on the pair of camera images. That is, for each camera of the two-camera model, K_{midpt} maps lists of four points in $\mathbb{P}^2$ to the intersection of the two lines defined by the four points.

Suppose that T is any given task function which can be factored as

$$T = T_{\text{new}} \circ H \tag{16}$$

where T_{new} is a new task function and H is a surjective function which factors through $\mathcal{C}$ for some K. Let f_{new} and y_{new} denote new feature and output vectors respectively, defined by the equations

$$f_{\text{new}} \triangleq H(f) \qquad\qquad y_{\text{new}} \triangleq K(y) \tag{17}$$

It is then easy to see that the tasks $T(f) = 0$ and $T_{\text{new}}(f_{\text{new}}) = 0$ are *equivalent* in the sense that

$$T(f) = 0 \iff T_{\text{new}}(f_{\text{new}})|_{f_{\text{new}}=H(f)} = 0, \quad \forall f \in \mathcal{F} \tag{18}$$

It is also easy to see that

$$y_{\text{new}} = \bar{C}_{\text{new}}(f_{\text{new}}) \tag{19}$$

where $\bar{C}_{\text{new}}$ is the extension of C_{actual} to $\mathcal{F}_{\text{new}}$.

Suppose that $E_{T_{\text{new}}}(y_{\text{new}}) = 0$ is an encoding that verifies $T_{\text{new}}(f_{\text{new}}) = 0$ on $\mathcal{C}$. Then, because of the surjectivity of H and the equivalence of $T_{\text{new}}(f_{\text{new}}) = 0$ and $T(f) = 0$ noted above, it must be true that verifiability of $T_{\text{new}}(f_{\text{new}}) = 0$ on $\mathcal{C}$ with $E_{T_{\text{new}}}(y_{\text{new}}) = 0$ implies verifiability of $T(f) = 0$ on $\mathcal{C}$ with $(E_{T_{\text{new}}} \circ K)(y) = 0$. The converse must also be true; that is, if $T(f) = 0$ is verifiable on $\mathcal{C}$ with the encoded task $(E_{T_{\text{new}}} \circ K)(y) = 0$, then $T_{\text{new}}(f_{\text{new}}) = 0$ is necessarily verifiable on $\mathcal{C}$ with the encoded task $E_{T_{\text{new}}}(y_{\text{new}}) = 0$.

Formally, a *modified Cartesian-based encoding* of a given task $T(f) = 0$ is then any encoding of the form

$$(T_{\text{new}} \circ \bar{C}^{-1}_{\text{est}_{\text{new}}} \circ K)(y) = 0$$

where T_{new} is a factor of T in a formula of the form $T = T_{\text{new}} \circ H$, H is a surjective function which factors through $\mathcal{C}$ with K, and $\bar{C}^{-1}_{\text{est}_{\text{new}}}$ is a left inverse of the extension of an estimate $C_{\text{est}} \in \mathcal{C}$ to $\mathcal{F}_{\text{new}}$. The form of a modified Cartesian-based task function is thus $E_T = T_{\text{new}} \circ \bar{C}^{-1}_{\text{est}_{\text{new}}} \circ K$. Note that

$$(T_{\text{new}} \circ \bar{C}^{-1}_{\text{est}_{\text{new}}})(y_{\text{new}}) = 0,$$

is a Cartesian-based encoding of $T_{\text{new}}(f_{\text{new}}) = 0$. Thus a modified Cartesian-based encoding of $T(f) = 0$ can be thought of as a transformed version of a Cartesian-based encoding of $T_{\text{new}}(f_{\text{new}}) = 0$.

Continuing with the example above, we can now define the task of determining if a fifth point is the midpoint of four coplanar points as

$$T_{\text{midpt}} = T_{\text{pp}} \circ H_{\text{midpt}}$$

and the corresponding task encoding as

$$E_{\text{midpt}} = T_{\text{midpt}} \circ \bar{C}^{-1}_{\text{est}_{\text{new}}} \circ K_{\text{midpt}}$$

It can be seen that a Cartesian-based encoding is a special case of a *modified* Cartesian-based encoding in which $\mathcal{F}_{\text{new}} = \mathcal{F}$, H is the identity on $\mathcal{F}$ and $T_{\text{new}} = T$. Thus every Cartesian-based encoding of a given task $T(f) = 0$ on $\mathcal{C}$ is a modified Cartesian-based encoding, whereas every modified Cartesian-based encoding is an image-based encoding. In other words, for any given class of two-camera models, we have the ordering

$$\mathcal{T}_{\text{Cbased}} \subset \mathcal{T}_{\text{MCbased}} \subset \mathcal{T}_{\text{Ibased}}$$

where $\mathcal{T}_{\text{MCbased}}$ denotes the class of all tasks which can be verified using modified Cartesian-based encodings.

4.3 Partial Calibration

In this section we introduce a new form of encoding, which is inspired by the Cartesian-based encoding, but that can be used to verify *any* decidable task. The key to achieve this property is to build into the encoding a form of online calibration—namely, a calibration that is consistent with the measured data.

Suppose that $T(f) = 0$ is a decidable task on some fixed set of two-camera models $\mathcal{C}$. Then the assignment

$$y \longmapsto \begin{cases} T\big(\bar{C}^{-1}(y)\big) & y \in \operatorname{Im} \bar{C} \text{ for some } f \in \mathcal{F} \text{ and } C \in \mathcal{C} \\ 1 & \text{otherwise} \end{cases} \tag{20}$$

where $\bar{C}^{-1}$ denotes any left inverse of $\bar{C}$, specifies a well-defined function $E_T : \mathcal{Y}^n \to \mathbb{R}$. In fact, the value of $E_T(y)$ computed by (20) is independent of the choice of $\bar{C}$ (provided that

$y \in \operatorname{Im} \bar{C}$) and the choice of its left inverse $\bar{C}^{-1}$. This is because, if there are two-camera models $C_1, C_2 \in \mathcal{C}$ such that $y \in \operatorname{Im} C_1 \cap \operatorname{Im} C_2$ and therefore $y = \bar{C}_1(f) = \bar{C}_2(g)$ for some $f, g \in \mathcal{F}$, then

$$T\big(\bar{C}_1^{-1}(y)\big) = T(f) = T(g) = T\big(\bar{C}_2^{-1}(y)\big)$$

Here we have used Lemma 2.

By construction, the encoding specified by (20) has the property that, for every $C \in \mathcal{C}$,

$$E_T(y)|_{y=\bar{C}(f)} = T(f), \quad f \in \mathcal{F}$$

Therefore, $E_T \circ \bar{C} = T,\ C \in \mathcal{C}$. From this and Lemma 1, one concludes that $T(f) = 0$ is verifiable on $\mathcal{C}$ with $E_T(y) = 0$. The following can then be stated:

Lemma 6 *If the task $T(f) = 0$ is decidable on some fixed set of two-camera models $\mathcal{C}$, then the encoding specified by* (20) *verifies $T(f) = 0$ on $\mathcal{C}$.*

What distinguishes the Cartesian-based encoding in (13) from the one specified by (20) is that in the latter, reconstruction is always done using a two-camera model that could have produced the measured data. This is because, in (20), for a given $y \in \mathcal{Y}$,

$$E_T(y) = T(f_{\text{new}})$$

with the estimated feature f_{new} given by

$$f_{\text{new}} = \bar{C}^{-1}(y)$$

for some two-camera model $C \in \mathcal{C}$ such that $y \in \operatorname{Im} \bar{C}$, i.e., for some two-camera model $C \in \mathcal{C}$ that could have produced the measurement y. In practice, this means that for different measurements in $\mathcal{Y}^n$ one may have to compute the estimated feature f_{new} using left inverses of distinct two-camera models. We can therefore regard the encoding specified by (20) as a form of Cartesian-based encoding that, for each measurement y, restricts the set of admissible two-camera models to those models that are compatible with y. In this sense, this encoding includes a form of online calibration. It is important to note that, however, that in many cases this calibration is only “partial” in the sense that the observed data may not be enough to perform even a weak calibration of the camera system.

4.4 Continuous Encodings

The results so far were mainly concerned with the existence of encodings, and little was mentioned regarding desirable properties of encodings, other than verifiability. A popular approach to developing a control algorithm is to view the encoded task function as a continuous error function which is driven to zero by a feedback control algorithm [37]. While our choice of encoded task functions, taking only the values 0 or 1, is often convenient for the preceding analysis, one might reasonably argue that, because of their discontinuous nature, such encodings have little practical application.

It turns out that under mild technical conditions on T and $\mathcal{C}$, and the assumption that $T(f) = 0$ is decidable, it is always possible to verify $T(f) = 0$ with an encoded task whose task function is continuous. To illustrate this, suppose that $\mathcal{V}$ is a compact subset of $\mathbb{R}^3$, that $\mathcal{Y}$ is $\mathbb{R}^2 \oplus \mathbb{R}^2$, that $||\cdot||$ is a norm on $\mathcal{Y}$, that $\mathcal{C}$ is of the form $\mathcal{C} = \{C_p : p \in \mathcal{P}\}$ where $\mathcal{P}$ a compact subset of a finite dimensional linear space, that $\{p, f\} \mapsto C_p(f)$ is a continuous function on $\mathcal{P} \times \mathcal{V}$, that the kernel of T is compact, and that $T(f) = 0$ is decidable on $\mathcal{C}$. Set

$$\mathcal{I} \triangleq C^*(\mathcal{P} \times \operatorname{Ker} T)$$

where $C^* : \mathcal{P} \times \mathcal{F} \to \mathcal{Y}^n$ is the function defined by the rule $(p, f) \longmapsto \bar{C}_p$. Consider the encoded task function $E_T : \mathcal{Y}^n \to \mathbb{R}$ defined so that

$$y \longmapsto \inf_{z \in \mathcal{I}} ||y - z|| \tag{21}$$

The following lemma states that E_T has the desired properties.

Lemma 7 *Under the above conditions, the task $T(f) = 0$ is verifiable on $\mathcal{C}$ with the encoding $E_T(y) = 0$. Moreover, E_T is globally Lipschitz continuous on $\mathcal{Y}$.*

This lemma is proved in the Appendix.

5 Concluding Remarks

The results of this paper can be summarized as follows. First, we have shown that

$$\mathcal{T}_{\text{pc}} \subset \mathcal{T}_{\text{uncal}} \subset \mathcal{T}_{\text{wkcal}} = \mathcal{T}_{\text{PInv}}$$

where the second subset relationship is strict. Second, we have shown that, for a given admissible class of two-camera models,

$$\mathcal{T}_{\text{pc}} \subset \mathcal{T}_{\text{Cbased}} \subset \mathcal{T}_{\text{MCbased}} \subset \mathcal{T}_{\text{Ibased}}.$$

Finally, for the specific case of weakly calibrated cameras, we have shown that

$$\mathcal{T}_{\text{Ibased}} = \mathcal{T}_{\text{PInv}}$$

Thus, we have established, in general, a "lower bound" on what tasks can be accomplished with two imprecisely modeled cameras, and in the specific case of weakly calibrated projective cameras, we have established an "upper bound" (and indeed, have shown that it can be attained). Furthermore, we have shown that the form of the encoding used to verify a task does affect what can be accomplished.

Practically speaking, one way to view these results are in terms of what they suggest about system design in the area of vision-based control. In particular, they provide a means to determine how much information is really needed about a camera system in order to accomplish a given task. In brief, some conclusions are:

- It is possible to verify that a robot positioning task has been accomplished with absolute accuracy using a weakly calibrated, noise-free, stereo vision system, if and only if the the task is invariant under projective transformations on $\mathbb{P}^3$. Thus, if a system designer can phrase the task in a ways which is projectively invariant, there must be a task-encoding which, given loop stability, will lead to accurate performance independent of the accuracy of the underlying system models.
- In many cases, it is possible to verify that such a robot positioning task has been precisely accomplished using an *uncalibrated* stereo vision system, thus obviating the need to rely even on an accurate "weak" calibration of the two-camera system.
- Even within the family of projectively invariant tasks, there is no need (and indeed a certain danger) to using a Cartesian-based control architecture.
- If such an architecture is to be used, the impact of calibration inaccuracy can be lessened or even eliminated through the use of modified Cartesian encodings, or some form of online calibration.

More generally, the issues addressed in this paper suggest a broader line of inquiry within the area of vision-based control. The central question would seem to be something like this: Given a set of one or more imprecisely modeled cameras and a task which might be positioning, tracking or something else, under what conditions can a task be accomplished with absolute precision using available images of features observed at one or more times? This question is, in a sense, concerned not with details of the specific image processing and control *algorithms* which might be used to accomplish the task, but rather with the *architecture* of a vision-based control system, the *structure* of its camera systems, and the level of *uncertainty* about that structure.

A related set of questions has to do with further characterization of task encodings and their relationship to robot tasks. For example, in developing a vision-based control system, it would be useful to define a "calculus" of tasks such that the definition of the task in turns defines the encoding for that task. The beginnings of such a calculus can be seen in the constructive results related to point-based tasks and projectively invariant tasks. Providing a complete, constructive characterization of projectively invariant tasks would thus form a design tool for practitioners in vision-based control.

Findings contributing to the answering of these and related questions should serve to strengthen our understanding of basic issues within the emerging field of computational vision and control.

Appendix

Proof of Lemma 2: Suppose $T(f) = 0$ is decidable. Then by Lemma 1, there must be an encoded task function E_T for which

$$\operatorname{Ker} T = \operatorname{Ker} E_T \circ \bar{C},\ C \in \mathcal{C} \tag{22}$$

Fix $C_1, C_2 \in \mathcal{C}$ and consider an arbitrary pair of feature lists $f, g \in \mathcal{F}$ such that $\bar{C}_1(f) = \bar{C}_2(g)$. Then $E_T \circ \bar{C}_1(f) = E_T \circ \bar{C}_2(g)$. From this and (22) it follows that $T(f) = 0$ if and only if $T(g) = 0$. Therefore $T(f) = T(g)$. Hence (6) holds.

Now suppose that (6) is true. Then $T(f) = T(g)$ whenever $C_1, C_2 \in \mathcal{C}$ and $f, g \in \mathcal{F}$ are such that $\bar{C}_1(f) = \bar{C}_2(g)$. Hence the assignment

$$y \longmapsto \begin{cases} T(f) & y = \bar{C}(f) \text{ for some } f \in \mathcal{F} \text{ and } C \in \mathcal{C} \\ 1 & \text{otherwise} \end{cases}$$

specifies a well-defined function $E_T : \mathcal{Y}^n \to \mathbb{R}$ which, for every $C \in \mathcal{C}$, satisfies

$$E_T(y)|_{y=\bar{C}(f)} = T(f),\ f \in \mathcal{F}$$

Therefore, $E_T \circ \bar{C} = T,\ C \in \mathcal{C}$. It follows that (22) is true. Thus by Lemma 1, $T(f) = 0$ is verifiable on $\mathcal{C}$ with $E_T(y) = 0$. Therefore $T(f) = 0$ is decidable. $\blacksquare$

Proof of Lemma 5: Let ℓ_1 be a projective line contained in $\mathcal{S} \cap A\mathcal{S}$, with $\mathcal{S} \triangleq \operatorname{span}\{e_1, e_2, e_3\}$ where each e_i denotes the ith column of the 4×4 identity matrix. Such an ℓ_1 exists because $\mathcal{S}$ and $A\mathcal{S}$ are two 3-dimensional linear subspaces of $\mathbb{R}^4$ and so their intersection is a linear subspace of $\mathbb{R}^4$ with dimension no smaller than 2. Note that no point in $\mathcal{V}$ is a linear subspace of $\mathcal{S}$ in $\mathbb{R}^4$.

Let ℓ be the baseline of G_0 and B a matrix in GL(4) such that $B\ell_1 = \ell$. We show next that $\mathbf{B}(\mathcal{V}) \subset \mathbb{P}^3_\ell$ and $\{\mathbf{B} \circ \mathbf{A}\}(\mathcal{V}) \subset \mathbb{P}^3_\ell$ and therefore that $C_1 \triangleq G_0 \circ (\mathbf{B}|\mathcal{V})$ and $C_2 \triangleq G_0 \circ (\{\mathbf{B} \circ \mathbf{A}\}|\mathcal{V})$ define two-camera models in $\mathcal{C}[G_0]$. This will finish the proof since $C_1 = G_1|\mathcal{V}$ and $C_2 = G_1 \circ (\mathbf{A}|\mathcal{V})$ with $G_1 \triangleq G_0 \circ (\mathbf{B}|\mathbb{P}^3_{\ell_1})$, because $\mathcal{V} \subset \mathbb{P}^3_{\ell_1}$. Suppose then that $\mathbf{B}(\mathcal{V})$ is not contained in $\mathbb{P}^3_\ell$ and therefore that there is a point $v \in \mathcal{V}$ such that $\mathbf{B}(v)$ is on ℓ. From this and the definition of B, v must be on ℓ_1 and therefore it must be a linear subspace of $\mathcal{S}$ in $\mathbb{R}^4$. This contradicts the fact that no point in $\mathcal{V}$ is a linear subspace of $\mathcal{S}$ in $\mathbb{R}^4$. To demonstrate that $\{\mathbf{B} \circ \mathbf{A}\}(\mathcal{V}) \subset \mathbb{P}^3_\ell$ we proceed similarly. By contradiction, suppose that $v \in \mathcal{V}$ and that $\mathbf{B}\big(\mathbf{A}(v)\big)$ is on ℓ. From this and the definition of B, $\mathbf{A}(v)$ must be on ℓ_1 and therefore it must be a linear subspace of $A\mathcal{S}$ in $\mathbb{R}^4$. Thus v must be a linear subspace of $\mathcal{S}$ in $\mathbb{R}^4$ and we arrive at a similar contradiction as before. $\blacksquare$

Proof of Lemma 7: We claim that $\mathcal{I}$ is compact. To prove that this is so, we first note that $\mathcal{P} \times \operatorname{Ker} T$ is compact because both $\mathcal{P}$ and $\operatorname{Ker} T$ are compact sets. Now the definition of C^* together with the assumed continuity of $\{p, f\} \mapsto C_p(f)$, implies that C^* is continuous.

Therefore, $\mathcal{I}$ must be compact since it is the image of a compact set under a continuous function.

To establish the Lipschitz continuity of E_T, let y_1 and y_2 be any two vectors in $\mathcal{Y}^n$. Assume without loss of generality that $E_T(y_1) \leq E_T(y_2)$. Since $\mathcal{I}$ is compact, $\inf_{z\in\mathcal{I}} ||y_1 - z||$ must be attained at some point $y^* \in \mathcal{I}$. In view of (21), $E_{T_{\text{new}}}(y_1) = ||y_1 - y^*||$ and $E_T(y_2) \leq ||y_2 - y^*||$. It follows that

$$|E_T(y_2) - E_T(y_1)| \leq ||y_2 - y^*|| - ||y_1 - y^*|| \leq ||y_2 - y_1||$$

Thus E_T is globally Lipschitz continuous as claimed.

In view of Lemma 1, to prove that $T(f) = 0$ is verifiable on $\mathcal{C}$ with $E_T(y) = 0$, it is sufficient to show that for each $p \in \mathcal{P}$,

$$\operatorname{Ker}(E_T \circ \bar{C}_p) = \operatorname{Ker} T$$

Toward this end first note that because $\mathcal{I}$ is a compact set, the kernel of E_T is exactly the set $\mathcal{I}$. Next note that because of the definitions of $\mathcal{I}$ and C^*, $\mathcal{I}$ can also be written as

$$\mathcal{I} = \bigcup_{p\in\mathcal{P}} \bar{C}_p(\operatorname{Ker} T) \tag{23}$$

Thus for any $f \in \operatorname{Ker} T$, it must be true that $\bar{C}_p(f) \in \mathcal{I},\ p \in \mathcal{P}$. But $\mathcal{I} = \operatorname{Ker} E_T$, so $\bar{C}_p(f) \in \operatorname{Ker} E_T,\ p \in \mathcal{P}$. This implies that $f \in \operatorname{Ker}(E_T \circ \bar{C}_p),\ p \in \mathcal{P}$ and thus that $\operatorname{Ker} T \subset \operatorname{Ker}(E_T \circ \bar{C}_p),\ p \in \mathcal{P}$.

For the reverse inclusion, fix $p \in \mathcal{P}$ and $f \in \operatorname{Ker}(E_T \circ \bar{C}_p)$. Then $\bar{C}_p(f) \in \operatorname{Ker} E_T$. Therefore $\bar{C}_p(f) \in \mathcal{I}$. In view of (23) there must be some $q \in \mathcal{P}$ and $g \in \operatorname{Ker} T$ such that $\bar{C}_p(f) = \bar{C}_q(g)$. Because of Lemma 2 this means that $T(f) = T(g)$. But $T(g) = 0$, so $T(f) = 0$ or equivalently $f \in \operatorname{Ker} T$. Hence $\operatorname{Ker}(E_T \circ \bar{C}_p) \subset \operatorname{Ker} T,\ p \in \mathcal{P}$. ■

Acknowledgment: The authors thank Radu Horaud and David Kriegman for providing encouragement and useful insights contributing to this work.

References

[1] B. Boufama, R. Mohr, and L. Morin. Using geometric properties for automatic object positioning. *Image and Vision Computing*, 16:27–33, 1998.

[2] A. Castano and S. A. Hutchinson. Visual compliance: Task-directed visual servo control. *IEEE Trans. Robot. Autom.*, 10(3):334–342, June 1994.

[3] W.-C. Chang. *Vision-Based Control of Uncertain Systems*. PhD thesis, Yale University, December 1997.

[4] W.-C. Chang, A. S. Morse, and G. D. Hager. A calibration-free, self-adjusting stereo visual control system. In *Proc. of the 13th World Congress, International Federation of Automatic Control*, volume A, pages 343–348. IFAC, July 1996.

[5] Wen-Chung Chang, João Pedro Hespanha, A. Stephen Morse, and Gregory D. Hager. Task re-encoding in vision-based control systems. In *Proc. of the 36th Conf. on Decision and Contr.*, pages 48–54, December 1997.

[6] F. Chaumette, E. Malis, and S. Boudet. 2d 1/2 visual servoing with respect to a planar object. In *Proc. IROS Workshop on New Trends in Image-based Robot Servoing*, pages 43–52, 1997.

[7] F. Chaumette, P. Rives, and B. Espiau. Classification and realization of the different vision-based tasks. In K. Hashimoto, editor, *Visual Servoing*, pages 199–228. World Scientific, 1994.

[8] P. Corke. Visual control of robot manipulators—a review. In K. Hashimoto, editor, *Visual Servoing*, pages 1–32. World Scientific, 1994.

[9] E. Dickmanns and V. Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1:241–261, 1988.

[10] B. Espiau, F. Chaumette, and P. Rives. A New Approach to Visual Servoing in Robotics. *IEEE Trans. Robot. Autom.*, 8:313–326, 1992.

[11] C. Fagerer, D. Dickmanns, and E. Dickmanns. Visual grasping with long delay time of a free floating object in orbit. *Autonomous Robots*, 1(1), 1994.

[12] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc., ECCV*, pages 563–578, 1992.

[13] O. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, Cambridge, MA, 1993.

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

[15] G. Hager. A modular system for robust hand-eye coordination. *IEEE Trans. Robot. Autom.*, 13(4):582–595, 1997.

[16] G. Hager, W.-C. Chang, and A. S. Morse. Robot hand-eye coordination based on stereo vision. *IEEE Control Systems Magazine*, 15(1):30–39, February 1995.

[17] G. Hager and Z. Dodds. A projective framework for constructing accurate hand-eye systems. In *Proceedings of the IEEE/RSJ/INRIA Workshop On New Trends in Image-Based Robot Servoing*, pages 71–82, 1997.

[18] R. I. Hartley. Self-calibration of stationary cameras. *Int. Journal of Computer Vision*, 22:5–24, 1997.

[19] R. I. Hartley. Chirality. *Int. Journal of Computer Vision*, 26(1):41–62, 1998.

[20] J. Hespanha. *Logic-Based Switching Algorithms in Control.* PhD thesis, Yale University, New Haven, CT, 1998.

[21] J. Hespanha, Z. Dodds, G. Hager, and A. S. Morse. Decidability of robot positioning tasks using stereo vision systems. To be presented at the 37th Conf. on Decision and Control, December 1998.

[22] J. P. Hespanha and A. S. Morse. Certainty equivalence implies detectability. In *Proc. NOLCOS*, June 1998.

[23] N. Hollinghurst and R. Cipolla. Uncalibrated stereo hand eye coordination. *Image and Vision Computing*, 12(3):187–192, 1994.

[24] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true jacobian. In *IEEE Int'l Workshop on Intelligent Robots and Systems*, pages 186–191. IEEE Computer Society Press, 1994.

[25] S. Hutchinson, G. Hager, and P. Corke. A tutorial introduction to visual servo control. *IEEE Trans. Robot. Autom.*, 12(5), 1996.

[26] M. Jagersand, O. Fuentes, and R. Nelson. Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *Proc., ICRA*, pages 2874–2880, 1997.

[27] R. Kelly. Robust asymptotically stable visual servoing of planar robots. *IEEE Trans. Robot. Autom.*, 12(5):759–766, October 1996.

[28] D. Kriegman, G. Hager, and A. S. Morse, editors. *The Confluence of Vision and Control.* Number 237 in Lecture Notes in Control and Information Sciences. Springer-Verlag, 1998.

[29] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[30] S. J. Maybank. Relation between 3d invariants and 2d invariants. *Image and Vision Computing*, 16:13–20, 1998.

[31] J. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, Mass., 1992.

[32] B.J. Nelson and P.K. Khosla. The resolvability ellipsoid for visual servoing. In *Proc. IEEE Conf. Comp. Vision and Patt. Recog.*, pages 829–832. IEEE Computer Society Press, 1994.

[33] J. Ponce and Y. Genc. A new approach to weak calibration. *Image and Vision Computing*, 16(3):223–243, 1998.

[34] A. Rizzi. *Dexterous Robot Manipulation.* PhD thesis, Yale University, November 1995.

[35] L. Robert and O. Faugeras. Relative 3D Positioning and 3D Convex Hull Computation from a Weakly Calibrated Stereo Pair. In *Proceedings of the International Conference on Computer Vision*, pages 540–543, Berlin, Germany, May 1993.

[36] L. Robert, C. Zeller, and O. Faugeras. Applications of non-metric vision to some visually guided robotics tasks. Technical Report 2584, INRIA, Sophia-Antipolis, June 1995.

[37] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: The Task Function Approach.* Clarendon Press, Oxford, England, 1992.

[38] Shankar Sastry and Mark Bodson. *Adaptive Control: Stability, Convergence and Robustness.* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[39] M. Seelinger, M. Robinson, Z. Dieck, and S. Skaar. A vision-guided, semi-autonomous system applied to a robotic coating application. In P. S. Schenker and G. T. McKee, editors, *Sensor Fusion and Decentralized Control in Autonomous Robotic Systems*, pages 133–144. SPIE, 1997.

[40] R. Sharma and S. Hutchinson. Motion perceptibility and its application to active vision-based servo control. *IEEE Trans. Robot. Autom.*, 13(1):61–73, February 1997.

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

[42] K. Toyama, J. Wang, and G. D. Hager. SERVOMATIC: a modular system for robust positioning using stereo visual servoing. In *Proc. IEEE Int'l Conf. Robot. and Automat.*, pages 2636–2643, 1996.

[43] L. Weiss, A. Sanderson, and C. P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE J. Robot. Automat.*, RA-3(5):404–417, October 1987.

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

[45] B. Yoshimi and P. K. Allen. Active, uncalibrated visual servoing. In *IEEE Int'l Conf. Robotics Automat.*, pages 156–161, San Diego, CA, May 1994.

[46] Z. Zhang. Determining epipolar geometry and its uncertainty: A review. *Int. Journal of Computer Vision*, 27(2):161–195, 1998.