Task Specification and Monitoring for Uncalibrated Hand/Eye Coordination*

Z. Dodds dodds@cs.yale.edu
G. D. Hager hager@cs.yale.edu
A. S. Morse morse@sysc.eng.yale.edu
Center for Computational Vision and Control
Yale University, New Haven, CT 06520

J. P. Hespanha hespanha@puccini.eecs.berkeley.edu
UCB Dept. of Electrical Eng.
Berkeley, CA 94720-1770

Abstract

Most of the work in robotic manipulation and visual servoing has emphasized *how* to specify and perform particular tasks. Recent results have formally shown *what* tasks are possible with uncalibrated imaging systems. This paper extends those results by characterizing in a constructive manner the set of tasks which can be performed with different types of uncalibrated camera models. The tasks' resulting structure provides a principled foundation both for a specification language and for automatic execution monitoring in uncalibrated environments. These two system components are prerequisites for extending existing feature-centered approaches to manipulation into a more powerful and intuitive object-centered architecture for hand/eye coordination.

1 Introduction

At a broad level, the goal of vision-guided robotics is to support both general and robust manipulation of *objects* using information acquired from images. Historically, however, much of the research in the area of vision-based control has focused on showing *how* very specific tasks can be achieved from *measured features* under visual control. In particular, a great deal of research has been devoted to developing feedback control systems which can accomplish specific positioning tasks without accurate estimates of camera calibration [1, 2, 6, 8, 14, 16, 25, 26, 27].

What is interesting in this case is the strong relationship between our knowledge about the underlying camera system and the structure of tasks which can be accomplished with precision. For example, it is clear that, given a perfectly calibrated hand/eye system, *any* task specified with respect to observed information can be accomplished with absolute precision. Yet in some cases tasks as simple as positioning to the midpoint of two observed points cannot be performed accurately if system calibration is not perfect.

This dichotomy led us to develop a a formal, complete characterization of *what* tasks can be accomplished under different camera model assumptions [12]. In this article, we extend these results in two ways:

- We provide a concise and *provably complete* task specification language for two families of uncalibrated two-camera systems and also derive a related family of image-based *task encodings* which can be used to implement vision-based control systems.
- We develop a canonical form which makes it possible to characterize the fundamental geometric structure of a given task.

The first characterization can be viewed as the basis for a feature-level "programming language" for manipulation systems. The second characterization is useful for implementing run-time monitoring of hand/eye task performance. We see both of these points as essential prerequisites for moving toward general, robust *object-level* manipulation systems.

The remainder of this article presents the formal results needed to develop these task characterizations and illustrate their use. Section 2 reviews a set of basic definitions. In particular, we define three families of "uncalibrated" camera models: injective, uncalibrated projective, and weakly calibrated cameras. Section 3 constructively characterizes the complete set of tasks performable with injective cameras and weakly calibrated cameras. Section 4 presents another characterization and shows how it supports task monitoring. Section 5 summarizes these results and how they contribute to the larger program of developing object-level manipulation systems.

Notation Throughout this paper, $'$ denotes matrix transpose, $\{e_i\}$ is the canonical basis of $\mathbb{R}^m$, and $\mathbb{P}^m$ is m-dimensional projective space. If a point $p \in \mathbb{P}^m$ is written in homogeneous coordinates, then norm(p)

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

is the same point with its coordinates scaled so that its last nonzero coordinate is a 1. $\mathcal{V}^n$ denotes the Cartesian product of a set $\mathcal{V}$ with itself n times. If $C : \mathcal{V} \to \mathcal{Y}$, we mildly abuse notation by continuing to use the function name C to denote the function mapping $\mathcal{V}^n$ to $\mathcal{Y}^n$ which uses C componentwise.

2 Background

In this paper we restrict attention to the interaction of point features within a robot's workspace $\mathcal{W}$. A stereo camera rig with a viewspace $\mathcal{V} \subset \mathcal{W}$ provides measurements of the image positions' of these features. We first describe precisely the camera models we are considering.

2.1 Camera Models

We denote by C_{actual} the actual stereo rig or *two-camera model* observing $\mathcal{W}$. We assume that the component cameras within C_{actual} have a joint field of view $\mathcal{V}$ containing an open subset of $\mathbb{P}^3$. C_{actual} then maps points from $\mathcal{V}$ to four-vectors of coordinates in the joint image space $\mathcal{Y} \subset \mathbb{P}^2 \times \mathbb{P}^2$ (the two image planes). Because we are not assuming a calibrated environment, C_{actual} is unknown. However, C_{actual} is assumed to be one of a known set, $\mathcal{C}$, of possible camera models.

We consider three general classes of two-camera models: injective cameras, uncalibrated projective cameras, and weakly calibrated projective cameras. Here, we summarize these models and refer to [12] for more details on how these classes are defined.

Let $\mathcal{C}_{\text{inj}}[\mathcal{V}]$ denote the class of all injective functions C such that $C : \mathcal{V} \to \mathcal{Y}$. This class is broad enough to include stereo rigs with lens distortions or panoramic cameras, as long as the cameras' baseline is outside $\mathcal{V}$.

A common and more geometrically meaningful case is the set of uncalibrated projective camera pairs. We define $\mathcal{C}_{\text{uncal}}[\mathcal{V}]$ to be the set of all projective two-camera models which are injective on $\mathcal{V}$. Thus, each component camera of a two-camera model in $\mathcal{C}_{\text{uncal}}[\mathcal{V}]$ can be modeled by a 3×4 matrix which maps points in $\mathcal{V}$ to points in $\mathbb{P}^2$ and is full-rank.[1]

It is possible from measured image information to compute a constraint on the parameters of an uncalibrated projective two-camera model. This constraint, the epipolar geometry, can be summarized by a fundamental matrix F [18]. An uncalibrated, projective two-camera model for which F is known is said to be *weakly calibrated* [9, 21]. We use the symbol $\mathcal{C}_{\text{wk}}[\mathcal{V}, F]$ to represent a set of two-camera models which are weakly calibrated, i.e., each two-camera model shares the same fundamental matrix F (up to a scale factor).

Another set of stereo rigs of interest are *weakly calibrated perspective* camera models, $\mathcal{C}_{\text{persp}}[\mathcal{V}, F, A]$. These are weakly calibrated projective models for which the internal camera parameters are known, i.e., the affine transformation A from pixel coordinates to camera-centered image coordinates is fixed.

In the sequel we will assume that the cameras' viewspace $\mathcal{V}$, an arbitrary fundamental matrix F, and a set of internal camera parameters A are understood and will suppress those parameters. We can then write relationships such as the following hierarchy:

$$C_{\text{actual}} \in \mathcal{C}_{\text{persp}} \subset \mathcal{C}_{\text{wk}} \subset \mathcal{C}_{\text{uncal}} \subset \mathcal{C}_{\text{inj}}. \qquad (1)$$

2.2 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" in $\mathcal{W}$. Both the pose of the robot under consideration and the target to which it is to be brought are determined by a list of simultaneously observed point features $\{f_1, f_2, \ldots, f_n\}$ in $\mathcal{V}$. We use an un-subscripted symbol such as f to denote each such list and we refer to f as a *feature*. The admissible feature space is the set $\mathcal{F} \stackrel{\triangle}{=} \mathcal{V}^n$ of all such lists of interest.

As in [16, 23], tasks are represented mathematically using a given *task function* T from $\mathcal{F}$ to $\{0, 1\}$. The *task* specified by T is the constraint

$$T(f) = 0. \qquad (2)$$

If (2) holds we say that the task is accomplished or satisfied at f. Note that this definition of a task function – operating on points in the robotic workspace – differs from some of the visual servoing literature [4, 5, 8, 10, 16, 22], in which task functions take image information as input. In addition, the codomain $\{0, 1\}$ abstracts the binary notion of satisfying or not satisfying a task specification, applications of which are this work's primary concern. In fact, only weak conditions on $\mathcal{V}$ and $\mathcal{Y}$ are needed to create a continuous task function in the sense of [8] from this abstraction [12].

2.3 Task Decidability

We seek to perform a task using information from cameras. As described in Section 2.1, the uncertainty in the actual camera configuration is represented by positing a set of two-camera models $\mathcal{C}$ within which the actual imaging system C_{actual} is known to lie. Thus, the information available to perform a task consists of the set $\mathcal{C}$, the task function T, and the measured data in the images:

$$y \stackrel{\triangle}{=} C_{\text{actual}}(f). \qquad (3)$$

If, on the basis of this information, it can be determined whether or not task (2) has been accomplished,

[1] For technical reasons discussed in [7], $\mathbb{P}^3 \setminus \mathcal{V}$ must include at least one plane (in $\mathbb{P}^3$). This is a very weak requirement, however, as the "plane at infinity" will never intersect $\mathcal{W}$.

then we say that the task is *decidable* on $\mathcal{C}$. To define decidability formally, we introduce the notion of an *encoded task function* [3] as a function $E_T : \mathcal{Y} \to \mathbb{R}$. The equation $E_T(y) = 0$ is then called an image-based *encoding* of the task $T(f) = 0$. This function of image data, E_T, closely resembles the prevailing use of the term "task function" in the literature.

We say that an encoding $E_T(y) = 0$ *verifies* a task $T(f) = 0$ on $\mathcal{C}$ if

$$\forall C \in \mathcal{C} \;\; \forall f \in \mathcal{F} \quad E_T(y) = T(f), \tag{4}$$

where $y = C(f)$. In turn, we say that a task is *decidable* on a set of camera models $\mathcal{C}$, if there exists some encoding which verifies that task on $\mathcal{C}$. Thus, the notion of decidability singles out precisely those tasks whose accomplishment (or lack thereof) can be deduced from measured data, without regard to particular encodings which verify them. Put another way, the essential characteristic of tasks decidable on $\mathcal{C}$ is the ability of the camera models in $\mathcal{C}$ to distinguish task success from task failure.

Using the notation $\mathcal{T}_{\mathrm{x}}$ to represent the set of tasks decidable on $\mathcal{C}_{\mathrm{x}}$, we can conclude from (1) and the universal quantifier in (4) that

$$\mathcal{T}_{\mathrm{inj}} \subset \mathcal{T}_{\mathrm{uncal}} \subset \mathcal{T}_{\mathrm{wk}} \subset \mathcal{T}_{\mathrm{persp}}. \tag{5}$$

2.4 Example Tasks

To make this formalism more concrete and provide a basis for our goal of characterizing decidable tasks in terms of primitive skills, we present several basic tasks:

Point-to-point task Let T_{pp} be the task function defined on $\mathcal{F}_{\mathrm{pp}} \triangleq \mathcal{V}^2$ by the rule

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

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

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

Midpoint Task Let T_{midpt} be the task function defined on $\mathcal{F}_{\mathrm{midpt}} \triangleq \mathcal{V}^3$ by the rule

$$\{f_1, f_2, f_3\} \longmapsto \begin{cases} 0 & f_1 \text{ is the midpoint of } f_2 \text{ and } f_3 \\ 1 & \text{otherwise} \end{cases}.$$

Coplanarity Task T_{copl} is a task function defined on $\mathcal{F}_{\mathrm{copl}} \triangleq \mathcal{V}^4$ by the rule

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

Cross-ratio Task Finally, let $T_{\mathrm{cr}\alpha}$ be the task function defined on $\mathcal{F}_{\mathrm{cr}} \triangleq \mathcal{V}^4$ by the rule

$$\{f_1, f_2, f_3, f_4\} \longmapsto \begin{cases} 0 & f_1, f_2, f_3, f_4 \text{ are collinear in } \mathbb{P}^3 \\ & \text{with cross ratio } \alpha \text{ [19]} \\ 1 & \text{otherwise} \end{cases}.$$

In fact, the set of tasks specifying point configurations with each possible cross-ratio $\alpha \in \mathbb{R}$ constitute a general ability to position metrically with respect to three collinear points; we term this set the *cross-ratio primitive.*

Some or all of these example tasks are a part of most implemented hand/eye systems; Section 3 pins down precisely the imaging systems required to ensure their decidability.

3 Characterizing Decidable Tasks

We employ the formalism of Section 2 to characterize completely the tasks decidable under different camera models. This characterization enables the specification of *any* decidable task using only a few task primitives and operators on those primitives. To be useful within this framework, operators on tasks must preserve decidability, and we next present a set of five such operators. We then demonstrate two sets of tasks spanned by these operators from a single primitive task.

3.1 Operators on Tasks

Let T, T_1, T_2 be task functions which take n-lists of feature points, i.e., $f \in \mathcal{V}^n$. We then define five operators on tasks and on task encodings, arranged in Figure 1.

With any operator on tasks, a natural first question is whether or not it preserves verifiability and thus decidability. The following propositions are straightforward to verify [7].

Proposition 1 *For any tasks T_1 and T_2, verified by E_{T_1} and E_{T_2} respectively, θE_{T_1} verifies θT_1 for each unary θ in Figure 1 and $E_{T_1} \theta E_{T_2}$ verifies $T_1 \theta T_2$ for each binary operator in Figure 1.*

Proposition 2 *The five operators of Figure 1 preserve task decidability on any set ($\mathcal{C}$) of two-camera models.*

We call the output (namely θT_1 or $T_1 \theta T_2$) of such operations *composite* task functions. This result allows us to proceed to generate tasks with the confidence that these composite tasks will not somehow "break" the imaging system's ability to distinguish success from failure.

Task Operators

- **Complement**
  Task: $\neg T(f) = 1 - T(f)$
  Encoding: $E_{\neg T}(y) = 1 - E_T(y)$
  provided T is a task
- **Permutation**
  Task: $\pi T(f) = T(\pi f)$
  Encoding: $E_{\pi T}(y) = E_T(\pi y)$
  with π a permutation of $\{1, 2, \ldots, n\}$
- **Disjunction**
  Task: $(T_1 \vee T_2)(f) = T_1(f)T_2(f)$
  Encoding: $E_{(T_1 \vee T_2)}(y) = E_{T_1}(y)E_{T_2}(y)$
  provided T_1 and T_2 are tasks
- **Expansion**
  Task: $\varepsilon T(\{g_1 \ldots g_m\}) = T(\{g_1 \ldots g_n\})$
  Encoding: $E_{\varepsilon T}(\{y_1 \ldots y_m\}) = E_T(\{y_1 \ldots y_n\})$
  with T a task and $n < m$
- **Contraction**
  Task: $cT(\{f_1 \ldots f_l\}) = T(\{f_1 \ldots f_n\})$
  Encoding: $E_{cT}(\{y_1 \ldots y_l\}) = E_T(\{y_1 \ldots y_n\})$
  with T a task, $l < n$, and cT well-defined

Figure 1: Five operators on tasks, along with corresponding encodings and conditions on the definitions.

3.2 Generating Tasks

The task hierarchy in (5) expresses the relative sizes of the sets of tasks decidable with different "uncalibrated" stereo rigs. This section grounds those relative relationships in terms of task families defined without recourse to camera models.

Point Coincidence Tasks - Injective Cameras The family $\mathcal{T}_{\rm pc}$ of *point-coincidence* tasks can be defined as the smallest set of tasks that contains the task $T_{\rm pp}(f) = 0$ and that is closed under task complement, permutation, disjunction, expansion, and contraction. In short, the family of point-coincidence tasks on $\mathcal{V}^n$ contains any task that can be fully specified by point-coincidence (or non-coincidence) relationships on n feature points. Point-coincidence tasks are the most commonly used specifications of visual servoing goals [4, 10, 15, 17].

To show that tasks in $\mathcal{T}_{\rm pc}$ are decidable on injective camera models, let $\mathcal{C}$ be an arbitrary set of injective 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

$$C_1(f) = C_2(g). \tag{6}$$

Suppose first that $T_{\rm pp}(f) = 0$ and therefore that $f_1 = f_2$. In this case (6) 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 $T_{\rm pc}(g) = 0$. Similarly one can conclude that $T_{\rm pp}(g) = 1$ whenever $T_{\rm pp}(f) = 1$. Thus, the task $T_{\rm pp}(f) = 0$ is decidable on $\mathcal{C}$. The following proposition then follows from this and Proposition 2.

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

Point-coincidence tasks are, in fact, the only tasks decidable on injective camera models, since for any other task it is possible to contrive an injective camera model on which success and failure are indistinguishible.

Projectively Invariant Tasks - Weakly Calibrated Cameras The following result [12] shows how additional knowledge about a hand/eye system's cameras can increase the set of tasks decidable by that system.

Proposition 4 *Let $\mathcal{C}_{\rm wk}$ be a set of injective, weakly calibrated camera models. A task $T(f) = 0$ is decidable on $\mathcal{C}_{\rm wk}$ if and only if it is projectively invariant.*

Here, a *projectively invariant task* is one which yields the same output for any two projectively equivalent inputs: if A is a projective transformation (extended to operate on lists of points) and $g = Af$, then f and g are *projectively equivalent* features and we write $f \simeq g$. Thus, the statement $T(g) = T(f) \Leftrightarrow f \simeq g$ characterizes projectively invariant tasks.

This result allows us to characterize tasks performable on weakly calibrated systems in a constructive manner similar to the point-coincident tasks above. In order to generate all decidable tasks on $\mathcal{C}_{\rm wk}$, it is necessary to span the set of projectively invariant tasks, denoted $\mathcal{T}_{\rm pi}$. In light of the fundamental role the cross-ratio plays in constructing projective invariants [19], the following proposition, proved in [7], combining the cross-ratio primitive with subspace-coincidence primitives, seems natural:

Proposition 5 *$\mathcal{T}_{\rm pi}$ is the closure of the cross-ratio primitive, coplanarity primitive, and point-to-point primitive under the operations depicted in Figure 1.*

Results (4) and (5) imply that, for example, the task $T_{\rm 3pt}$ is decidable when weak calibration is known. However, they do not show an image-based encoding (on which a controller might be built) which verifies this task. The following section extends these characterizations to task encodings in a constructive manner.

3.3 Building Image-based Encodings of Tasks

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 $E_{\mathrm{pp}}(y)$ on $\mathcal{Y}^n$ by the rule[2]

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

It is simple to show that $E_{\mathrm{pp}}(y) = 0$ verifies $T_{\mathrm{pp}}(f) = 0$ on $\mathcal{C}_{\mathrm{inj}}$.

Using the operations on encodings demonstrated in Figure 1, one can obtain an encoding that verifies a given point-coincidence task $T(f) = 0$ on $\mathcal{C}_{\mathrm{inj}}$ by applying the same construction to the encoding that was used to build the task $T(f) = 0$ from the primitive $T_{\mathrm{pp}}(f) = 0$. This gives a systematic procedure for building encodings that verify any point-coincidence task on $\mathcal{C}_{\mathrm{inj}}$.

Similarly, define $E_{\mathrm{cr}\alpha}(y)$ by

$$\{y_1, y_2, y_3, y_4\} \mapsto \begin{cases} 0 & y_1^j, y_2^j, y_3^j, y_4^j \text{ are collinear and} \\ & \text{have cross ratio } \alpha \ (j = 1, 2) \\ 1 & \text{otherwise} \end{cases},$$

with $y_i = [y_i^1 \ y_i^2]$; $y_i^j \in \mathbb{R}^2$ are the individual projected features in each of the two images. Then, $E_{\mathrm{cr}\alpha}(y) = 0$ verifies the task $T_{\mathrm{cr}\alpha}(f) = 0$. In principle, encodings of projectively invariant tasks can then be built from $E_{\mathrm{cr}\alpha}(y) = 0$ and the operators in Section 3.1.

It is important to note that, while this result guarantees completeness, it does not restrict a system designer to combinations of cross-ratios for specifying tasks in $\mathcal{T}_{\mathrm{wk}}$. In fact, encodings of the other primitive tasks of Section 2.4, provided they are decidable on $\mathcal{C}_{\mathrm{wk}}$, would certainly be used to provide a more intuitive interface for task construction. A task-specification language based on encodings for T_{pp} and T_{3pt}, as well as $E_{\mathrm{cr}\alpha}$, would generate all possible projectively invariant tasks. T_{midpt}, however, which is not decidable on $\mathcal{C}_{\mathrm{wk}}$, could not be used as a building block within weakly calibrated systems.

The weakly calibrated perspective camera models $\mathcal{C}_{\mathrm{persp}}$ mentioned in Section 2.1 can decide additional tasks not in $\mathcal{T}_{\mathrm{pi}}$. For example, the midpoint-positioning task, T_{midpt} is decidable on $\mathcal{C}_{\mathrm{persp}}$ [7], but it is shown in the next section that $T_{\mathrm{midpt}} \notin \mathcal{T}_{\mathrm{wk}}$.

3.4 Summary

The results of this section have provided a constructive approach to characterizing the task hierarchy of (5):

$$\mathcal{T}_{\mathrm{pc}} = \mathcal{T}_{\mathrm{inj}} \subset \mathcal{T}_{\mathrm{uncal}} \subset \mathcal{T}_{\mathrm{wk}} = \mathcal{T}_{\mathrm{pi}} \subset \mathcal{T}_{\mathrm{persp}}.$$

Although we do not have a complete characterization of $\mathcal{T}_{\mathrm{uncal}}$, we do have, in effect, "bounds" on the capabilities of systems with uncalibrated projective cameras.

Propositions 3 and 5 yield a concise and provably complete language for specifying the tasks performable by a hand/eye system with injective and weakly calibrated imaging systems, respectively. This provides a basis for constructing a programming interface to such systems. In addition, these characterizations allow a system to compose task encoding functions into encodings for composite tasks.

4 Geometric Properties of $\mathcal{T}_{\mathrm{wk}}$

The fact that weakly calibrated hand/eye systems can decide projectively invariant tasks suggests a connection between projective coordinate systems and performable tasks. This section exploits this connection. In particular, we show how to use the structure of $\mathcal{T}_{\mathrm{wk}}$ in order to evaluate a system's capabilities even when the feature points are lost or occluded visually. The result is the basis for an automatic task monitor which balances between a task's need for information and a vision system's ability to provide it.

4.1 Canonical Representatives of Elementary Tasks

Put another way, (4) states that the task $T(f) = 0$ is decidable on $\mathcal{C}_{\mathrm{wk}}$ exactly when T is constant on each equivalence class of $\mathcal{V}^n$ (under projective equivalence of feature lists). It can be shown that as long as $\mathcal{V}^n$ is an open set in $(\mathbb{P}^3)^n$, the set of equivalence classes of $\mathcal{V}^n$ is identical to the equivalence classes of $(\mathbb{P}^3)^n$ [13]. Thus, we can investigate the structure of the set $\mathcal{T}_{\mathrm{wk}}$ by examining the set of equivalence classes of $(\mathbb{P}^3)^n$. A natural subset of $\mathcal{T}_{\mathrm{wk}}$ to consider are those tasks satisfied on exactly one feature equivalence class. We call these *elementary tasks*.

Each equivalence class of $(\mathbb{P}^3)^n$ (or elementary task) can be represented by a "simplest" list of features, which we take to be that list in the class whose feature points' homogeneous coordinates consist of as many zeros, and then ones, as possible. We insist on this for each feature point, in order from left to right, and its coordinates, written from bottom to top. Each resulting *canonical representative* of an equivalence class of $(\mathbb{P}^3)^n$ exposes the projective structure of the individual points comprising its feature list.

For concreteness, consider four feature lists $f^1 \ldots f^4$ for which $n = 3$:

$$\underbrace{\begin{bmatrix} 0 & 1 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}}_{f^1} \underbrace{\begin{bmatrix} 5 & 2 & 8 \\ 0 & 0 & 0 \\ 3 & 6 & 0 \\ 1 & 1 & 1 \end{bmatrix}}_{f^2} \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{f^3} \underbrace{\begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}}_{f^4}.$$

[2] This encoding, taking values in $\{0, 1\}$, does not convey enough information to control a manipulator. The natural encoding $E_{\mathrm{pp}}(\{y_1, y_2\}) = ||y_1 - y_2||$ does suffice, however, as shown in numerous implemented examples [16]. We use binary-valued encodings only to simplify the presentation here (e.g., in defining the negation operator).

Assuming the usual embedding of $\mathbb{R}^3$ within $\mathbb{P}^3$, the points in f^1 are three points along a single axis in $\mathbb{R}^3$, and those in f^2 satisfy the midpoint task T_{midpt}. Both are projectively equivalent to the canonical representative f^3 which expresses that its three feature points lie in a single line and no two are coincident. Put another way, f^3 can be considered the elementary task which is satisfied exactly when three points are collinear but none coincident. Because T_{midpt} is satisfied for feature f^2 but not f^1 and both features are projectively equivalent to f^3, Proposition 4 implies that T_{midpt} is *not* decidable on a weakly calibrated set of cameras.

4.2 Properties of the Equivalence Classes

One fundamental attribute of features $f \in (\mathbb{P}^3)^n$ is the number of dimensions spanned by their consituent feature points when considered vectors in $\mathbb{R}^4$. We use $\dim(f)$ to denote this quantity. Thus, $1 \leq \dim(f) \leq 4$, as long as there is at least one feature point in the list. For the features f^1, f^2, and f^3, $\dim(f^i) = 2$. Because projective transformations are invertible, they preserve dimensionality, so that $\dim(f)$ is a constant for all features in a given equivalence class of $(\mathbb{P}^3)^n$. From this, we can conclude that since $\dim(f^4) = 3$ and $\dim(f^3) = 2$, f^4 and f^3 are members of distinct equivalence classes of $(\mathbb{P}^3)^n$, i.e., they represent two distinct task specifications decidable on C_{wk}.

A complete list of the elementary tasks, then, reduces to a list of canonical representatives for the equivalence classes of $(\mathbb{P}^3)^n$. The following rules provide a means for generating all such representatives of size n from those of size $n-1$.

If $f^- = [f_1 \ldots f_{n-1}] \in (\mathbb{P}^3)^{n-1}$ is in canonical form, then $f = [f_1 \ldots f_n] \in (\mathbb{P}^3)^n$ is in canonical form if and only if f_n is a *legal successor* of f^-, i.e., f_n satisfies one of the following properties:

1. **Coinciding** $f_n = f_i$ for some $1 \leq i < n$. In this case the new feature point coincides with one of the previous ones.

2. **Broaching** $f_n = e_i$, where $i = \dim(f^-) < 4$. In this case the spanning dimension of f is larger than that of f^-, i.e., the new feature point introduces another dimension. When a "1" appears in a row of a feature, that row (dimension) has been "broached."

3. **Measuring** $f_n = s + e_i$ $(1 \leq i \leq \dim(f^-)$ and $s \cdot e_i = 0)$, where s is a legal successor of f^-. In this case f_n creates a projective coordinate system within the dimensions broached by s and e_i. In effect, this rule adds a second "1" to one or more previously broached dimensions. Because arbitrary positioning is then possible in these dimensions, they are considered "measured."

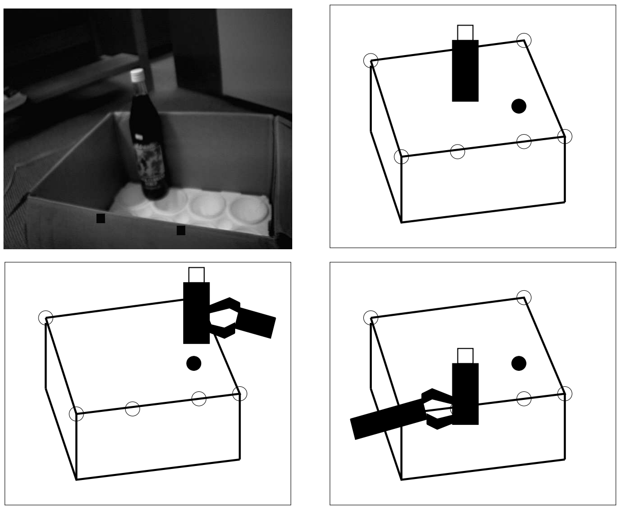

Figure 2: (**Top Left**) An example box-packing task. (**Top Right**) A stylized version of the task specification. The dark circle represents the goal point for the bottle's base. Open circles represent reference points with known relative geometry. (**Lower Left**) Under partial feature occlusion the task may not be possible without additional features, or (**Lower Right**) may remain possible, depending on the remaining points' configuration.

4. **Positioning** $f_n = \text{norm}(v)$, where $v \in \mathbb{R}^4, v \neq 0$ contains any real numbers in those dimensions measured by f^- and zeros elsewhere. In this case the point f_n is being positioned to a unique location within the coordinate system defined by the points in f^-.

A proof of the validity of this set of rules appears in [7], resulting in the following proposition.

Proposition 6 *If a feature f^- is in canonical form, then the feature $f = \{f^-; f_n\}$ is also in canonical form if and only if f_n is a legal successor of f^- as defined above.*

Starting from the degenerate task (the always-satisfied task of size $n = 1$, whose representative is $[1\ 0\ 0\ 0]'$), these rules provide a method for constructing all of the elementary tasks point by point. Furthermore, all tasks are disjunctions of some subset of elementary tasks.

In essence, these rules present constraints on what kinds of feature points can be specified within a previously known set of points. If we interpret the latter as a projective coordinate system, these constraints determine the extent to which an additional feature point can be positioned in that coordinate system.

4.3 Monitoring Task Execution

Given an existing projective coordinate system, the rules of Proposition 6 offer a complete characterization of the positioning tasks possible with an "additional" point (or points). Thus, a task monitor – a process keeping track of the visual information available for a task – can use these rules to distinguish between those situations in which lack of visual information prohibits performing a task and other cases when occlusion or tracking failure need not affect performance. An example is the box-packing task depicted in Figure 2, in which the goal is to position the bottom of a bottle at a particular spot (the filled circles) above a box before performing an insertion.

The coordinates of the box's feature points (counterclockwise, in cm) and the canonical representative of this projective coordinate system (the open circles in the upper right frame), respectively, are

$$\begin{bmatrix} 0 & 0 & 15 & 30 & 43 & 43 \\ 34 & 0 & 0 & 0 & 0 & 34 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \simeq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & \alpha & \beta \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where α and β represent the projectively invariant coordinates of the last two points, written in terms of the coordinate system of the preceding feature points. Because any point of the form $[\gamma\ \delta\ 1\ 0]'$ is a legal successor of the canonical representative for this six-point projective coordinate system, arbitrary positioning in the plane formed by those six points is possible.

If, on the other hand, a task-monitoring process detects that some of the box's feature points have been occluded, it can regenerate the canonical form for the remaining visible points in order to determine the effect of the occlusion on the task at hand. For example, under the occlusions shown in the lower left and lower right frames of Figure 2, respectively, the canonical description of the visible points are

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & \alpha \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since $[\gamma\ \delta\ 1\ 0]'$ is not a legal successor in the first case but is in the second, the task monitor can proceed with the task in the latter, but must direct the vision system to seek additional known features in the former situation.

It is well-known that four planar points are required to perform metrical positioning with respect to those points in the plane – with the caveat that no three of those points may be collinear. This guideline and others governing projective coordinate systems are concisely represented by Proposition 6. By maintaining the list of visible object features, converting to canonical form, and employing these rules, a task-monitoring process can decide to continue with a task or to perform another routine to search for additional feature points.

5 Conclusions

This paper characterizes the tasks performable with injective cameras and with a weakly calibrated vision system. The result is a concise specification language for decidable tasks in terms of a small set of primitive skills and task operators. A second characterization of elementary performable tasks leads to a set of rules for constructing projective coordinate systems. These rules provide an organizational framework for determining when the available visual information suffices to proceed with task execution.

Within this framework, several related questions remain unanswered. An accuracy analysis, augmenting the decidability analysis of Sections 3 and 4 and perhaps along the lines of [20, 24], would allow a task-monitoring process to make decisions based on the tolerance within which a task must be performed. Also, characterizations of the tasks decidable on sets of camera models other than injective and weakly calibrated are less well understood.

Most important, perhaps, is the issue of incorporating the capabilities for task specification and analysis presented here into a higher-level framework. Most existing hand/eye systems present a feature-centered interface in which a user must detail the interactions among visual primitives. We feel that the characterizations of performable tasks presented here provide two components required by a more user-friendly, object-centered system: a simple, complete language for specifying tasks and a means for automatically overseeing task execution. Figure 3 presents the skeleton of such a system. Hand/eye control primitives and vision primitives exist; this paper presents a principled basis for constructing a user interface for specifying tasks and automated monitoring. The remaining components – object models and system integration – form the basis of ongoing work.

References

[1] P.K. Allen, B. Yoshimi, and A. Timcenko. Hand-eye coordination for robotics tracking and grasping. In K. Hashimoto, editor, *Visual Servoing*, pages 33–70. World Scientific, 1994.

[2] A. Castaño 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, J. P. Hespanha, A.S. Morse, and G.D. Hager. Task re-encoding in vision-based control systems. In *Proceedings, Conference on Design and Control*, San Diego, CA, December 1997.

[4] 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.

[5] 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.

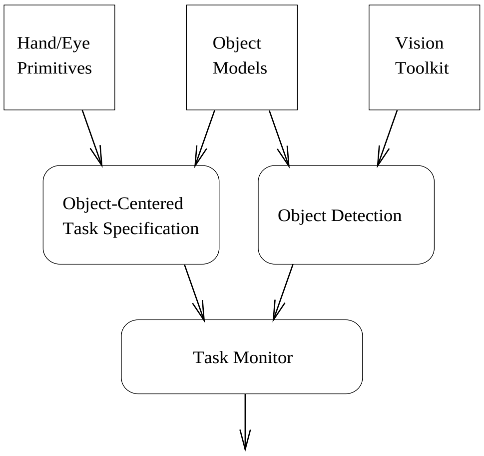

Figure 3: **System Skeleton** of an object-centered, rather than a feature-centered hand/eye workcell. The characterizations of decidable tasks presented in this paper support a principled approach to the user interface (specifying tasks) and the task monitoring (evaluating visual conditions) components of the system.

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

[7] Z. Dodds, G. D. Hager, J. Hespanha, and A. S. Morse. On the structure of tasks decidable on uncalibrated stereo rigs. Technical Report TR-1138, Yale University, New Haven, CT, 1998.

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

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

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

[11] G. D. Hager and Z. Dodds. A projective framework for constructing accurate hand-eye systems. In *Proc. IROS Workshop on New Trends in Image-based Robot Servoing*, pages 71–82, 1997.

[12] J. Hespanha, Z. Dodds, G. D. Hager, and A. S. Morse. What can be done with an uncalibrated stereo system? In D. J. Kriegman, G. D. Hager, and A. S. Morse, editors, *The Confluence of Vision and Control*, number 237 in Lecture Notes in Control and Information Sciences, pages 79–89. Springer-Verlag, London, 1998.

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

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

[15] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true jacobian. pages 186–191. IEEE Computer Society Press, 1994.

[16] S. A. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. Robot. Automat.*, 12(5):651–670, October 1996.

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

[18] Q. Luong and O. D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17:43–75, 1996.

[19] J. L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, MA, 1992.

[20] B. J. Nelson and P. K. Khosla. The resolvability ellipsoid for visual servoing. pages 829–832. IEEE Computer Society Press, 1994.

[21] L. Robert and O. D. 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.

[22] 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.

[23] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: The Task Function Approach.* Number 22 in The Oxford engineering science series. Clarendon Press, Oxford, 1991.

[24] 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.

[25] 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.

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

[27] 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.