Issues Concerned with Task Encoding in Vision-Based Control Systems

J. P. Hespanha, W-C. Chang, A. S. Morse, G. D. Hager
Center for Computational Vision and Control
Yale University

Feedback control systems employing video cameras as sensors have been studied in the robotics community for many years. An especially interesting feature of such systems is that *both* the process output {e.g., the position of the robot in its workspace} and the reference set-point {e.g., landmark determined target} can be and often are observed through the *same* sensors {i.e., cameras}. Because of this unusual architectural feature, it is sometimes possible to achieve *precise* positioning {at least in the absence of measurement noise}, despite not only process model imprecision {as in the case of a conventional set-point control system with a perfect loop-integrator and precise output and exogenous reference sensing} but sensor imprecision as well. But in contrast to a setpoint control system where what to choose for an error is usually pretty clear, in vision-based systems there prove to be many choices for errors, with different attributes.

Some of the observations just made are implicit in work extending back more than 15 years [1]. Some of these issues are touched upon in the recent tutorial on visual servoing [2], and in the monograph by [3]. The concept of task encoding is discussed briefly in the Doctoral Thesis of [4] which contains further references.

Some of the preceding issues have been discussed quite recently in [5] which considers the problem of controlling the position of a rigid robot in a prescribed *workspace* $\mathcal{W} \subset \mathbb{R}^3$ using data observed by two cameras functioning as a stereo vision system. The observed data consists of the robot position in $\mathcal{W}$ as well as various geometrical features of interest which appear in the cameras' fields of view $\mathcal{V} \subset \mathbb{R}^3$. Invariably $\mathcal{W} \subset \mathcal{V}$.

In [5] the robot is assumed to admit a simple kinematic model of the form $\dot{x} = u$ where x is a state vector whose components are the Cartesian coordinates of the robot in $\mathcal{W}$, and u is a control vector taking values in $\mathcal{W}$. The robot's position in $\mathcal{W}$ as well as various features of interest in $\mathcal{V}$ {e.g., a line segment, a circle, etc.} are seen in an *image space* $\mathcal{Y} \triangleq \mathbb{R}^2 \oplus \mathbb{R}^2$ through a fixed but imprecisely known readout function or *camera model* $C_{\text{actual}} : \mathcal{V} \to \mathcal{Y}$ which describes a two camera, stereo vision system. Thus the robot's position in image space is a measured output y related to x by the formula $y = C_{\text{actual}}(x)$. Correspondingly, a specific *feature* {i.e., subset} f in $\mathcal{V}$, is seen in $\mathcal{Y}$ as $z = C_{\text{actual}}(f)$.

Generally speaking one would need C_{actual} to be at least injective in order for the vision subsystem to provide the requisite information for positioning. On the other hand, one would never expect to know precisely what C_{actual} is. Accordingly, [5] assumes that C_{actual} is a fixed but unknown element of a prescribed set of injective functions $\mathcal{C}$ which each map $\mathcal{V}$ into $\mathcal{Y}$. Depending on the specific visual servoing problem under consideration, $\mathcal{C}$ might be a finite set, a neighborhood of C_{actual} in a metric space, a finite collection of disjoint neighborhoods or something else.

A typical positioning problem or *task* consists of driving a robot to a 'target' {or set} in $\mathcal{W}$ determined by a list of observed features {i.e., subsets} in $\mathcal{V}$. In practice, target sets are determined not by just one observed feature $f_1 \in \bar{\mathcal{V}}$, the power set of $\mathcal{V}$, but rather by an ordered set of n simultaneously observed features $\{f_1, f_2, \ldots, f_n\} \in \bar{\mathcal{V}}^n$ where each f_i is a member of some specific class of like or similar objects in $\bar{\mathcal{V}}$ and $\bar{\mathcal{V}}^n$ is the n-fold Cartesian product

$$\bar{\mathcal{V}}^n \stackrel{\Delta}{=} \underbrace{\bar{\mathcal{V}} \times \bar{\mathcal{V}} \times \ldots \times \bar{\mathcal{V}}}_{n \text{ times}}$$

For example, f_1 might be a member of a set of line segments in $\mathcal{V}$, f_2 a member of a set of cubes in $\mathcal{V}$, etc.

Let $\mathcal{F}$ be a given subset feature lists in $\bar{\mathcal{V}}$. The relationship between observed lists of features $f \stackrel{\Delta}{=} \{f_1, f_2, \ldots, f_n\}$ in $\mathcal{F}$ and the target in $\mathcal{W}$ each such list determines, is typically characterized by a given function $T : \mathcal{F} \to \bar{\mathcal{V}}$ called a *target map*; invariable $T(\mathcal{F}) \subset \mathcal{W}$. Each list of features $f \in \mathcal{F}$ specifies a target set $T(f) \subset \mathcal{W}$ to which the robot is to be moved. In other words, for a given feature $f = \{f_1, f_2, \ldots, f_n\} \in \mathcal{F}$, the *task* to be accomplished is to cause the robot state to satisfy

$$\boxed{x \in T(f)} \tag{1}$$

Subject to the assumption that C_{actual} is in $\mathcal{C}$ but is otherwise unknown, the *control problem* is to develop a causal algorithm which accomplishes (1), using only sensed data $y = C_{\text{actual}}(x)$ and $z = \bar{C}_{\text{actual}}(f)$ where for each $C \in \mathcal{C}$, $\bar{C}$ denotes the injective function $\bar{C} : \bar{\mathcal{V}} \to \bar{\mathcal{Y}}^n$,

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

$$\bar{\mathcal{Y}}^n \stackrel{\Delta}{=} \underbrace{\bar{\mathcal{Y}} \times \bar{\mathcal{Y}} \times \ldots \times \bar{\mathcal{Y}}}_{n \text{ times}}$$

and $\bar{\mathcal{Y}}$ is the power set of $\mathcal{Y}$.

For given $x \in \mathcal{W}$ and given $f \in \mathcal{F}$ the question "Is (1) true?" is called a *task query*. The family of all possible task queries

$$\mathcal{Q} \stackrel{\Delta}{=} \{\text{Is } x \in T(f)? : x \in \mathcal{W},\ f \in \mathcal{F}\}$$

is said to be *observable on* $\mathcal{C}$ if there exists an algorithm $\mathfrak{A}$ which, for each $C \in \mathcal{C}$, each $x \in \mathcal{W}$ and each $f \in \mathcal{F}$, is able to decide whether or not (1) is true from the measured data $y = C(x)$ and $z = C(f)$.

An explicit set of necessary and sufficient conditions has recently been derived which characterizes those families of task queries which are observable. Meanwhile in [5] are discussed three different ways to "encode" (1) into implementable tasks which if achieved, imply that the original task (1) is achieved as well. The widely used *Cartesian-based* encoding is motivated by "certainty equivalence." In the present context, certainty equivalence advocates that one should use estimates $\widehat{x}$ and $\widehat{f}$ of x and f respectively to accomplish task (1), with the understanding that these estimates are to be viewed as correct even though they may not be. Since $y = C_{\text{actual}}(x)$ and $z = \bar{C}_{\text{actual}}(f)$, it is natural to choose as these estimates $\widehat{x} \stackrel{\Delta}{=} C^{-1}(y)$ and $\widehat{f} \stackrel{\Delta}{=} \bar{C}^{-1}(z)$, where C^{-1} and $\bar{C}^{-1}$ are left inverses of C and $\bar{C}$ respectively. The Cartesian-based encoding of (1) would thus be

$$\widehat{x} \in T(\widehat{f}) \tag{2}$$

In [5], a partial answer is given to the following question: When does achieving the re-encoded task (2) imply that the original task (1) has been achieved as well? It turns out that point-to-point positioning with the target point and the measured feature being one and the same, can always be accomplished with this encoding. On the other hand, something as simple as moving a robot to a line in $\mathcal{W}$ determined by two feature points in $\mathcal{V}$ cannot be achieved with Cartesian-based encoding.

Also examined in [5] are two other encoding methods, namely *modified Cartesian-based* and *image-based*. The image-based encoding of (1) is a task of the form $y \in \bar{T}(z)$ where, if it exists, $\bar{T}$ is a well-defined function of $T, \mathcal{C}$, and $\mathcal{F}$. If $\bar{T}$ exists, then satisfaction of the image-based encoding $y \in \bar{T}(z)$ implies satisfaction of the original encoding (1). It is shown in [5] that for the point-to-line task mentioned above, such a $\bar{T}$ does indeed exist. Thus for the point-to-line task, the image-based encoding does the job but the Cartesian-based encoding does not. Despite this and other more general advances made along these lines, at present the basic issue of task encoding is far from settled. Just how one might encode a particular task and go about defining the appropriate error signals to be zeroed is not typically clear, especially when the target is determined by many observed features or when the task to be accomplished is something more challenging than simple positioning.

References

[1] A. C. Sanderson and L. E. Weiss. Image-based visual servo control using relational graph error signals. In *Proc. of IEEE*, pages 1074–1077, 1980.

[2] S. A. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE J. of Robotics and Automation*, 12(5):651–670, October 1996.

[3] 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.

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

[5] W C Chang, J. P. Hespanha, A. S. Morse, and G. D. Hager. Task re-encoding in vision-based control systems. Technical report, Yale University, March 1997.