Submitted, 2010 International Conference on Robotics and Automation (ICRA)
http://www.cds.caltech.edu/~murray/papers/hcsm10-icra.html

# A bootstrappable bio-plausible design for visual pose stabilization

Shuo Han, Andrea Censi, Andrew D. Straw, and Richard M. Murray

***Abstract*—We consider the problem of purely visual pose stabilization of a second-order rigid-body system: how to choose forces and torques, based on the visual input alone, such that the view converges to a memorized goal image. Emphasis has been given to the bio-plausibility of the computation, in the sense that the control laws could be in principle implemented on the neural substrate of simple insects. We show that stabilizing laws can be realized by bilinear/quadratic operations on the visual input. Moreover, the control laws can be "bootstrapped" (learned unsupervisedly) from experience, which further substantiate the bio-plausibility of such computation.**

## I. Introduction

The question of how the brain utilizes millions of neurons, presumably in a highly parallel fashion, to make timely decisions from information-rich sensory inputs still remains unanswered. This remains true even for one of the simplest animals, the fruit fly, which has only about 300,000 neurons, orders of magnitude fewer than the human brain. Engineers marvel at the series of *fast* and *robust* behaviors implemented on the poor computational support of *slow* and *noisy* neurons. Particularly interesting is the processing of visual information (Fig. 1), which biologists believe is a dominant sense controlling flight in many species of insects. Examples of classical studied behaviors include turning towards features of interest [1] and altitude/velocity control [2]. In this paper, we are especially interested in the visual behaviors that allow insects to return to or to stay at a particular location in space (*hovering* and *homing*). Leading theories posit that navigation is performed using a combination of panoramic retinal snapshots from key locations taken during learning flights [3], [4] and distance information relating such locations to each other [5]–[8]. Such local representations are tied to prominent local landmarks, and insects may navigate by traveling from landmark to landmark using these local maps, dead reckoning, or a compass cue such as the pattern of polarization in the sky [9], [10]. Alternative views of bee navigation propose that maps are formed [11] or that visual-motor control loops are strengthened during learning [12].

The hovering and homing behaviors are instances of what in robotics is called *visual servoing*. Our goal in this paper is to solve this problem while respecting the constraints of a "bio-plausible" computation. The motivation is twofold: on the one hand, this might provide clues for realizing a computational model of the insects behavior, useful to biologists; on the other hand, it might inspire useful engineering principles to realize similar robust systems. The question of what can be considered bio-plausible, is, of course, loosely defined. In brief, we consider something to be bio-plausible if it can be realized on neural circuits [13]. The computation should be implementable using parallel asynchronous units; in insects, a model of computation that is well accepted as bio-plausible is a local nonlinear operation followed by wide-field linear integration.

The authors are with the Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA. {hanshuo, astraw}@caltech.edu, {andrea, murray}@cds.caltech.edu

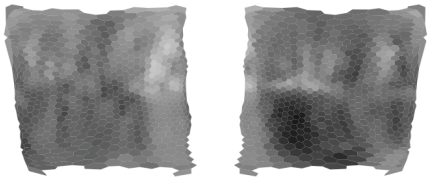
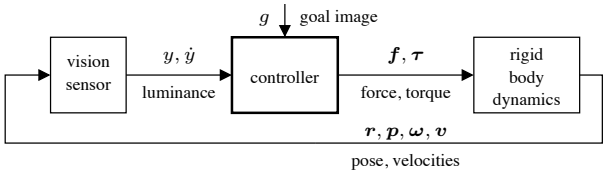
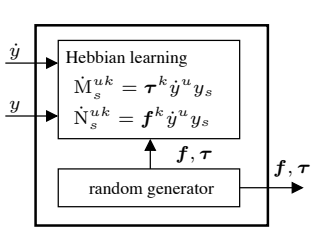


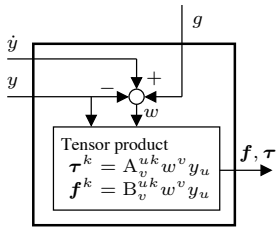
Figure 1. Example of simulated visual input for a fruit fly. The 1,398 ommatidia of the two compound eyes cover almost all the visual sphere.




(a) Model for purely visual pose stabilization.




(b) Hebbian learning of control tensors.




(c) Control as a tensor product.

Figure 2. (a) We control in force/torque a rigid body model based only on the luminance profile. (b) We bootstrap the behavior by forcing the system with random input and performing Hebbian learning. (c) Proportional-derivative pose stabilization control laws are realized as one local nonlinearity and a tensor product (wide-field integration). Refer to the text for notations.

An example of computation that could not be considered bio-plausible would be feature-based visual servoing: not as much for implementing a feature-extraction algorithm, but because the logic of manipulating a "list of features" and matching them does not appear to be implementable on a neural substrate. The alternative is using the raw values of pixel luminance, which, interestingly, is a new trend in robotics [14]–[17]. The feasibility of this scheme for insects has been considered, and an experimental assessment [18] concluded that "view-based homing with panoramic images is in principle feasible in natural environments and does not require the identification of individual landmarks".

In this paper, we present proportional-derivative control

laws that can solve the visual servoing problem for a second-order system controlled in force/torque. We do not rely on velocity sensors; we only assume to know the current visual observations (Fig. 2a). The computation essentially consists of a tensor product of the goal image and the current observations (Fig. 2c). Moreover, our control laws are *bootstrappable* from experience: the control tensor can be learned unsupervisedly using Hebbian learning during a calibration phase (Fig. 2b). This is interesting in biology, because it allows us to postulate that the genotype only encodes a generic learning algorithm rather than extensive pre-wired information. From the engineering point of view, this allows one to work with a completely uncalibrated camera, in which each pixel is arbitrarily oriented on the visual sphere.

*Related work:* Visual servoing is a classical problem in industrial robotics, with applications in controlling robot manipulators [19], [20] or visual navigation [21], [22]. The classical approach normally consists of extracting and tracking fiducial point/segments from the image [23]. However, in the last few years, several methods have been proposed that do not rely on features, but instead use the image intensity directly [14]–[17]. These are formulated as computing gradient flows of an image dissimilarity measure; the obvious but not unique choice is the squared error over the field of view. An issue that appears not to be completely solved at the moment is formulating sufficient conditions for convergence, especially because these gradient flows depend on the distance to the objects, which is usually approximated.

In feature-based visual servoing, learning ("adaptive") techniques have been used for inferring the kinematics (image Jacobian) [24], [25] and for action planning [26]. This learning, however, appears to be strongly structured and relies on precomputations such as feature extraction. Actually, the learning/bootstrapping part of this paper has been inspired by the somewhat unrelated research of Kuipers on the *spatial semantic hierarchy* [27], which provides a framework for bootstrapping successive abstractions of the world, from uninterpreted sensors to cognitive maps; the learning presented in this paper corresponds to the lower levels of the hierarchy, applied to the vision sensor.

*Contribution:* This paper is the ideal continuation of [28], where we solved the same problem for purely rotational motion, and proposed a *supervised* learning algorithm (back-propagation) to learn the control tensor. We give two main improvements over the previous results: 1) We solve the problem for rotation + translation instead of pure rotation. This is not a trivial extension, because when performing translation, the visual observation will be affected by the distance to objects, which is not instantaneously observable. 2) We show that the control law can be *bootstrapped* via an *unsupervised* Hebbian learning rule, which makes the computation more likely to be implemented on a neural substrate.

With respect to related work in engineering on feature-free visual servoing, we offer the following contributions:

- We consider a second-order system: the control inputs are torque/force rather than velocities; this implies estimating velocity from variations in the luminance to create suitable damping laws.
- We prove the stability of purely visual control laws independent of (or robust to changes in) the distance profile; previous work did propose laws assuming constant distance, but did not prove convergence (e.g., [16]).
- The aforementioned learning part, from the engineering point of view, allows us to obtain a controller that requires completely zero configuration: it does not rely on the geometry of the camera, because the pixels are considered merely as an unordered numerical array.

The paper is organized as follows. Section II introduces the notation used and preliminaries about the model. Section III considers about the observability of the problem. Section IV–VI show, in the order of complexity: 1) damping control laws that stabilize the velocities to zero, 2) control laws for the first order system, controlled in velocity, and 3) proportional-derivative control for the second-order system. Section VII shows how these control laws can be bootstrapped. Section VIII shows simulations that validate the correctness of the strategy using the simulated visual input of a fruit fly.

## II. Preliminaries

This paper uses a basic differential-geometric language for describing motions; the reader not acquainted with such notation can refer to, e.g., [29]. For example, $D_x f(x)|_{x=x_0} \cdot v$ refers to the derivative of $f$ with respect to $x$ applied to the tangent vector $v$ at $x = x_0$. This reduces to the directional derivative when $f$ is defined on a Euclidean space.

Let $\mathrm{SO}(3)$ ($\mathrm{SE}(3)$) be the special orthogonal (Euclidean) group of dimension 3. Let $\mathbb{S}^2 \subset \mathbb{R}^3$ be the unit sphere. In the following, "$s$" always signifies an element of $\mathbb{S}^2$. For example, we model visual input as a function $y(s,t)$, where $t$ is time, and $s$ spans the visual sphere $\mathbb{S}^2$. For such a function, we assume the convention that $\nabla_s y(s,t)$, or $\nabla y$ for short, is an element of $\mathbb{R}^3$; i.e., we think of it as an "arrow attached to the sphere $\mathbb{S}^2$". Many computations involve integration over the sphere: $\langle f \rangle \triangleq \int f(s)\, d\mathcal{S}$, with $\mathcal{S}$ as the unique rotation-invariant measure on $\mathbb{S}^2$.

We denote the algebraic dual of a vector space $\mathbf{E}$ as $\mathbf{E}^*$. In the finite-dimensional case, $\boldsymbol{x}^* \in \mathbf{E}^*$ reduces to the vector transpose: $\boldsymbol{x}^*(\boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$ ($\forall\, \boldsymbol{y} \in \mathbf{E}$). The tensor products on $\mathbf{E}$ and $\mathbf{E}^*$ are described using the Einstein convention: $y^i \in \mathbf{E}$, $y_i = \left(y^i\right)^* \in \mathbf{E}^*$, $\mathrm{M}_{ij} \in L^2(\mathbf{E}; \mathbb{R})$ (bilinear forms). During multiplication, the repeated indices are summed or integrated over: $y_i g^i = \sum_i y_i g^i$ ($i = 1, \cdots, n$), $y_s g^s = \langle yg \rangle$ ($s \in \mathbb{S}^2$).

*System dynamics:* We consider the fully actuated rigid body dynamics on $\mathrm{SE}(3)$. The state space consists of the body attitude $\boldsymbol{r} \in \mathrm{SO}(3)$, the body position $\boldsymbol{p} \in \mathbb{R}^3$, the angular velocity in the body frame $\boldsymbol{\omega} \in \mathbb{R}^3$, and the linear velocity in the body frame $\boldsymbol{v} \in \mathbb{R}^3$. The dynamics can be written as

$$\begin{cases} \dot{\boldsymbol{r}} &= \boldsymbol{r}\,(\boldsymbol{\omega})^\wedge, \\ \mathbb{I}\dot{\boldsymbol{\omega}} &= (\mathbb{I}\boldsymbol{\omega}) \times \boldsymbol{\omega} - \epsilon_{\boldsymbol{\omega}} \boldsymbol{\omega} + \boldsymbol{\tau}, \\ \dot{\boldsymbol{p}} &= \boldsymbol{r}\boldsymbol{v}, \\ m\dot{\boldsymbol{v}} &= m\boldsymbol{v} \times \boldsymbol{\omega} - \epsilon_{\boldsymbol{v}} \boldsymbol{v} + \boldsymbol{f}, \end{cases} \tag{1}$$

where $(\cdot)^\wedge$ is the "hat map" [30] that maps a vector $\boldsymbol{x}$ to the corresponding skew-symmetric matrix $\boldsymbol{x}^\wedge$ such that $\boldsymbol{x}^\wedge \boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y}$, with "$\times$" being the ordinary cross product in $\mathbb{R}^3$; $\mathbb{I} \in \mathbb{R}^{3\times 3}$ is the momentum of inertia; $m \in \mathbb{R}$ is the body mass; $\boldsymbol{\tau} \in \mathbb{R}^3$ is the control torque; $\boldsymbol{f} \in \mathbb{R}^3$ is the control force. We also include two damping terms ($-\epsilon_{\boldsymbol{v}}\boldsymbol{v}$, $-\epsilon_{\boldsymbol{\omega}}\boldsymbol{\omega}$) to make the model more realistic. Occasionally, we will also refer to the first-order system, assuming we can set $\boldsymbol{\omega}$ and $\boldsymbol{v}$ directly.

*Sensors:* We assume to have available only the output of a vision sensor. With good generality, a vision sensor is a device that at each time $t$ returns an array of $n$ values $y^i(t)$, each corresponding to the light intensity in a direction $s^i \in \mathbb{S}^2$, convolved by a certain point-spread function. This formulation is equally apt for modelling a normal camera, a catadioptric camera, or the ommatidia of a fruit fly. Later on, it will be clear that we do not even need to know each direction $s^i$; i.e., our sensor is completely uncalibrated.

In the pure rotation case, the value of $y(s,t)$ can be written as $y(s,t) = m(\boldsymbol{r}(t)s)$, where $m(s)$ is the luminance profile in the reference attitude. However, in the presence of translation, it is not possible to give a closed-form expression of $y(s,t)$ as a function of $\boldsymbol{r}$ and $\boldsymbol{p}$, because it depends, in a complicated way, on the "nearness" $\mu(s,t)$, which is the inverse of the distance to the object in direction $s$. We only need the relation between $\dot{y}(s,t)$ and $\mu(s,t)$, recalled in Lemma 1 in the appendix.

In deriving the theory, we make a number of simplifying assumptions on the vision sensor: 1) We assume that we can sample the entire visual field and that we have "a continuum of pixels", i.e., we know $y(s,t)$ for all $s \in \mathbb{S}^2$; 2) We assume that we can observe both $y(t)$ and $\dot{y}(t)$; 3) We ignore the effect of occlusions. Note, however, that in the simulations we do incorporate blurring, sensor dynamics, and occlusions.

*Control problem:* We assume that the agent knows a "goal" image, $g(s)$, taken at the goal pose $\boldsymbol{q}_g = (\boldsymbol{r}_g, \boldsymbol{p}_g)$. Without loss of generality, we often assume that $(\boldsymbol{r}_g, \boldsymbol{p}_g) = (\mathrm{Id}, 0)$. The problem we wish to solve can be stated as follows:

*Problem 1 (Visual pose stabilization):* Given the goal image $g$, design a stabilizing control law for $\boldsymbol{\tau}$ and $\boldsymbol{f}$, depending only on $y$, $\dot{y}$, $g$, such that $\boldsymbol{q} \to \boldsymbol{q}_g$.

## III. OBSERVABILITY AND CONTRAST

The first issue is assessing whether knowing only $y$ and $g$ is sufficient to drive the system to the goal. In other words, we should check whether $\boldsymbol{q}_g$ is observable from the observations $y$. One way is checking that $\boldsymbol{q} = \boldsymbol{q}_g$ is the only solution of the equation $y(s) = g(s)$. Equivalently, $\boldsymbol{q} = \boldsymbol{q}_g$ should be an isolated minimum of the distance in image space:

$$J(\boldsymbol{q}) = \frac{1}{2}\left\langle (y-g)^2 \right\rangle. \tag{2}$$

$J$ depends on $\boldsymbol{q}$ because $y$ varies with $\boldsymbol{q}$ . A sufficient condition for $\boldsymbol{q}_g$ to be observable is that the Hessian of $J$ is positive definite at $\boldsymbol{q}_g$. Here we stumble on a technical difficulty: the attitude $\boldsymbol{r}$ lives on the manifold $\mathrm{SO}(3)$; therefore the Hessian is not simply a "square matrix" as in standard calculus. See the text [31] for the subtleties involved in defining the Hessian of a function on a manifold. To circumvent the difficulty, we make a local change of coordinates from $\mathrm{SO}(3)$ to $\mathbb{R}^3$, for which standard calculus can be applied. In particular, we choose the exponential coordinates $\boldsymbol{\varphi} \in \mathbb{R}^3$:

$$\boldsymbol{\varphi} = \mathrm{Log}(\boldsymbol{r})^\vee, \qquad \boldsymbol{r} = \mathrm{Exp}(\boldsymbol{\varphi}^\wedge),$$

where Log and Exp are the *matrix* logarithm and exponential, and $(\cdot)^\vee$ is the inverse of the hat map. Note in particular that $(\boldsymbol{r} = \mathrm{Id}) \Leftrightarrow (\boldsymbol{\varphi} = 0)$.

*Proposition 1:* The Hessian of $J(\boldsymbol{\varphi}, \boldsymbol{p})$, also referred to as the "contrast matrix", is

$$\mathrm{C}(y,\mu) = \begin{bmatrix} \langle \mathrm{S}y \mathrm{S}y^* \rangle & \langle \mu \mathrm{S}y \nabla y^* \rangle \\ \langle \mu \nabla y \mathrm{S}y^* \rangle & \langle \mu^2 \nabla y \nabla y^* \rangle \end{bmatrix}, \tag{3}$$

where $y(s)$ is the image, $\mu(s)$ is the nearness profile, $\nabla = \nabla_s$, and the operator S is defined as

$$\mathrm{S}y \triangleq s \times \nabla_s y. \tag{4}$$

*Proof:* The result is a direct consequence of Lemma 2 and 3 in the appendix. ∎

From now on, we assume the following condition:

*Condition 1 (Full contrast):* The matrix $\mathrm{C}(g, \mu_g)$ is positive definite.

If the condition is not satisfied, the environment possesses some kind of spatial invariance that makes Problem 1 impossible to solve with the visual input alone[1]. Analogously, we define a global version of the contrast condition, as follows.

*Condition 2 (Global full contrast):* At all poses, the matrix $\mathrm{C}(y,\mu)$ is positive definite.

## IV. VISUAL DAMPING CONTROL LAWS

Our ultimate goal is constructing a proportional-derivative controller. In this section, we restrict the focus on how to create a *damping* control law: choose force and torque as a function of $y$, $\dot{y}$, such that the velocities $\boldsymbol{\omega}$, $\boldsymbol{v}$ are driven to 0, even in the absence of the natural damping factors in (1).

### A. Bilinear estimation/damping of angular velocity.

We have shown in [28] that a bilinear (BL) form of $y$, $\dot{y}$, defined as $\hat{\boldsymbol{\omega}}_{\mathrm{BL}} \triangleq \langle (\mathrm{S}y)\, \dot{y} \rangle$ , suffices to estimate the rotational velocity "up to 90 deg": $\boldsymbol{\omega}^* \hat{\boldsymbol{\omega}}_{\mathrm{BL}} \geq 0$. This is a velocity estimator that is *arbitrarily inaccurate* in scale, without more constraints on the environment. In fact, suppose that $\hat{\boldsymbol{\omega}}_{\mathrm{BL}} = \boldsymbol{\omega}$ in a certain environment $y$. For another environment that is twice as bright: $y' = 2y$, the resulting $\hat{\boldsymbol{\omega}}_{\mathrm{BL}}$ will be off by a factor of 4. Nevertheless, we showed that $\hat{\boldsymbol{\omega}}_{\mathrm{BL}}$ is useful for control purposes:

*Proposition 2 (Pure rotation damping):* Assume that Condition 2 holds and $\boldsymbol{v} = 0$. Then the torque command $\boldsymbol{\tau} = -k_d \langle (\mathrm{S}y)\, \dot{y} \rangle$ stabilizes the angular velocity $\boldsymbol{\omega}$ to 0.

[1]Excluding degenerate cases in which the Hessian is singular but higher-order derivatives are not.

### B. Bilinear estimation/damping of linear velocity.

We consider the analogous results for the linear velocity, assuming pure translational motion ($\boldsymbol{\omega} = 0$). Define the bilinear estimator of $\boldsymbol{v}$ as $\hat{\boldsymbol{v}}_{\mathrm{BL}} \triangleq \langle (\nabla y)\, \dot{y} \rangle$, and analogous properties to the rotation case can be proved:

*Proposition 3 (Pure translation damping):* Assume Condition 2 holds and $\boldsymbol{\omega} = 0$. Then $\boldsymbol{v}^* \hat{\boldsymbol{v}}_{\mathrm{BL}} \geq 0$. Moreover, the force command $\boldsymbol{f} = -k_d \langle (\nabla y)\, \dot{y} \rangle$ stabilizes the linear velocity $\boldsymbol{v}$ to 0.

*Proof:* Condition 2 implies that $\left\langle \mu^2 \nabla y \nabla y^* \right\rangle > 0$, hence $\langle \mu \nabla y \nabla y^* \rangle > 0$ since $\mu \geq 0$. Thus, by Lemma 1 in the appendix, $\boldsymbol{v}^* \hat{\boldsymbol{v}}_{\mathrm{BL}} = \boldsymbol{v}^* \langle \dot{y}\, (\nabla_s y) \rangle = \boldsymbol{v}^* \langle \mu \nabla y \nabla y^* \rangle \boldsymbol{v} > 0$. The dynamics of $\boldsymbol{v}$, given by $\dot{\boldsymbol{v}} = m^{-1} \boldsymbol{f} = -k_d m^{-1} \langle (\nabla y)\, \dot{y} \rangle = -k_d m^{-1} \langle \mu \nabla y \nabla y^* \rangle \boldsymbol{v}$, are asymptotically stable. ∎

### C. Joint velocity damping.

We have shown that a damping control law can be created from bilinear functions of $\dot{y}$ and $y$, in the cases of pure rotation and pure translation. Does this mean that the joint damping law works? This is not immediate, because the dynamics of $\boldsymbol{\omega}$ and $\boldsymbol{v}$ interact. We can prove this only in the presence of a modified contrast condition, which will be important also for the next sections.

For $\alpha > 0$, define the matrix

$$\tilde{\mathrm{C}}(y, \mu, \alpha) \triangleq \begin{bmatrix} \langle \mathrm{S} y \mathrm{S} y^* \rangle & \left\langle \left(\frac{\alpha}{2} + \frac{\mu}{2}\right) \mathrm{S} y \nabla y^* \right\rangle \\ \left\langle \left(\frac{\alpha}{2} + \frac{\mu}{2}\right) \nabla y \mathrm{S} y^* \right\rangle & \alpha \langle \mu \nabla y \nabla y^* \rangle \end{bmatrix}.$$

*Condition 3 (Modified full contrast):* There exists $\alpha > 0$ such that $\tilde{\mathrm{C}}(g, \mu_g, \alpha)$ is positive definite.

*Remark 1:* The need for this condition stems from the fact that $\tilde{\mathrm{C}}(y, \mu, \alpha) = \frac{1}{2}(\mathrm{A} + \mathrm{A}^*)$, where the matrix

$$\mathrm{A} \triangleq \begin{bmatrix} \langle \mathrm{S} y \mathrm{S} y^* \rangle & \alpha \langle \nabla y \mathrm{S} y^* \rangle \\ \langle \mu \nabla y \mathrm{S} y^* \rangle & \alpha \langle \mu \nabla y \nabla y^* \rangle \end{bmatrix} \tag{5}$$

will appear in many of the proofs. As of now, we do not have an intuitive interpretation of the matrix $\tilde{\mathrm{C}}(y, \mu, \alpha)$, except in one special case. Suppose the environment is spherical: $\mu(s) = \overline{\mu}$, and let $\alpha = \overline{\mu}$. In this case, $\tilde{\mathrm{C}}(y, \mu, \alpha) = \mathrm{C}(y, \mu)$, thus Condition 3 is equivalent to Condition 1. By continuity, we can also argue that if $\mathrm{C}(y, \mu) > 0$, then $\tilde{\mathrm{C}}(y, \mu, \alpha) > 0$ (and vice versa) if the environment is of sufficiently small variation from spherical. We conjecture this condition could be equivalent to the full contrast condition, at least in the case of $\mu > 0$ (no objects at infinity).

With this condition, it is straightforward to show that the joint damping law is stable.

*Proposition 4 (Joint damping):* Assume Condition 3 holds uniformly at every point ($\tilde{\mathrm{C}}(y, \mu, \alpha) > 0$). Then the control law ($k_d > 0$)

$$\begin{cases} \boldsymbol{\tau} = -k_d \langle (\mathrm{S} y) \dot{y} \rangle, \\ \boldsymbol{f} = -\alpha k_d \langle (\nabla y) \dot{y} \rangle \end{cases}$$

stabilizes $\boldsymbol{v}, \boldsymbol{\omega}$ to 0.

*Proof:* Take the energy $V(\boldsymbol{\omega}, \boldsymbol{v}) = \frac{1}{2} m \boldsymbol{v}^* \boldsymbol{v} + \frac{1}{2} \boldsymbol{\omega}^* \mathbb{I} \boldsymbol{\omega}$ as a Lyapunov candidate, and compute the derivative of $V$:

$$\begin{aligned} \dot{V} &= \boldsymbol{v}^* m \dot{\boldsymbol{v}} + \boldsymbol{\omega}^* \mathbb{I} \dot{\boldsymbol{\omega}} \\ &= \boldsymbol{v}^* (m \boldsymbol{v} \times \boldsymbol{\omega} - \epsilon_{\boldsymbol{v}} \boldsymbol{v} + \boldsymbol{f}) + \boldsymbol{\omega}^* ((\mathbb{I}\boldsymbol{\omega}) \times \boldsymbol{\omega} - \epsilon_{\boldsymbol{\omega}} \boldsymbol{\omega} + \boldsymbol{\tau}) \\ &= -\epsilon_{\boldsymbol{v}} \boldsymbol{v}^* \boldsymbol{v} + \boldsymbol{v}^* \boldsymbol{f} - \epsilon_{\boldsymbol{\omega}} \boldsymbol{\omega}^* \boldsymbol{\omega} + \boldsymbol{\omega}^* \boldsymbol{\tau} \\ &\leq \boldsymbol{v}^* \boldsymbol{f} + \boldsymbol{\omega}^* \boldsymbol{\tau} \\ &= \boldsymbol{v}^* (-\alpha k_d \langle (\nabla y) \dot{y} \rangle) + \boldsymbol{\omega}^* (-k_d \langle (\mathrm{S} y) \dot{y} \rangle). \end{aligned}$$

Now consider that from Lemma 1,

$$\begin{aligned} \langle (\nabla y) \dot{y} \rangle &= \langle \mu \nabla y \nabla y^* \rangle \boldsymbol{v} + \langle \nabla y \mathrm{S} y^* \rangle \boldsymbol{\omega}, \\ \langle (\mathrm{S} y) \dot{y} \rangle &= \langle \mu \mathrm{S} y \nabla y^* \rangle \boldsymbol{v} + \langle \mathrm{S} y \mathrm{S} y^* \rangle \boldsymbol{\omega}. \end{aligned}$$

Thus $\dot{V}$ is a quadratic function of $(\boldsymbol{v}, \boldsymbol{\omega})$:

$$\begin{aligned} \dot{V} &= -k_d \left[ \alpha \boldsymbol{v}^* \left( \langle \mu \nabla y \nabla y^* \rangle \boldsymbol{v} + \langle \nabla y \mathrm{S} y^* \rangle \boldsymbol{\omega} \right) \right. \\ &\qquad \left. + \boldsymbol{\omega}^* \left( \langle \mu \mathrm{S} y \nabla y^* \rangle \boldsymbol{v} + \langle \mathrm{S} y \mathrm{S} y^* \rangle \boldsymbol{\omega} \right) \right] \\ &= -k_d \left[ \begin{array}{cc} \boldsymbol{\omega} & \boldsymbol{v} \end{array} \right] \mathrm{A} \left[ \begin{array}{cc} \boldsymbol{\omega} & \boldsymbol{v} \end{array} \right]^*. \end{aligned}$$

If Condition 3 holds, $\dot{V}$ is negative definite by Remark 1, and the system converges to $\boldsymbol{v}, \boldsymbol{\omega} = 0$. ∎

## V. Control in velocity

We now consider the problem of controlling the first-order system, assuming that we can impose $\boldsymbol{\omega}$ and $\boldsymbol{v}$ directly. A natural way to define control laws is expressing them as *gradient flows* of the quadratic cost function (2).

*Proposition 5 (Exact gradient flow):* The gradient flow that minimizes (2) is

$$\begin{cases} \boldsymbol{\omega} = \langle (\mathrm{S} y)(g - y) \rangle, \\ \boldsymbol{v} = \langle \mu (\nabla y)(g - y) \rangle. \end{cases} \tag{6}$$

*Proof:* Compute $\dot{J}$ as $\dot{J} = \langle (y - g) \dot{y} \rangle$ and substitute the expression for $\dot{y}$ in Lemma 1 (in the appendix). ∎

Note that, similarly to the damping control law of the previous section, these gradient flows have a particularly simple computational form, being bilinear in the error $(g - y)$ and $y$. Unfortunately, $\boldsymbol{v}$ depends on the nearness profile $\mu(s)$, which is unknown.

One possible solution would be estimating $\mu$, by solving a *structure from motion* problem (e.g., [32]–[34]). Our approach, instead, is to investigate whether it is possible to use a control law that does not depend on $\mu$. Serendipitously, we could just drop $\mu$ and hope it would work. In fact, it does, as explained by the following proposition:

*Proposition 6 (Approximated gradient flow):* Suppose Condition 3 holds. Then the control law

$$\begin{cases} \boldsymbol{\omega} = \langle (\mathrm{S} y)(g - y) \rangle, \\ \boldsymbol{v} = \alpha \langle (\nabla y)(g - y) \rangle \end{cases} \tag{7}$$

locally stabilizes the first-order system.

*Proof:* Consider the first-order dynamical system as in (1) and rewrite it using exponential coordinates as

$$\begin{cases} \dot{\boldsymbol{\varphi}} = \boldsymbol{\omega}, \\ \dot{\boldsymbol{p}} = \mathrm{Exp}(\boldsymbol{\varphi}^\wedge) \boldsymbol{v}. \end{cases}$$

The linearization around the equilibrium $(\boldsymbol{\varphi}, \boldsymbol{p}) = (0, 0)$ is

$$\begin{cases} \dot{\boldsymbol{\varphi}} = D_{\boldsymbol{\varphi}}(\boldsymbol{\omega}) \cdot \boldsymbol{\varphi} + D_{\boldsymbol{p}}(\boldsymbol{\omega}) \cdot \boldsymbol{p}, \\ \dot{\boldsymbol{p}} = D_{\boldsymbol{\varphi}}(\boldsymbol{v}) \cdot \boldsymbol{\varphi} + D_{\boldsymbol{p}}(\boldsymbol{v}) \cdot \boldsymbol{p}. \end{cases}$$

Using Lemma 2 and 3, and recurring again to the definition of A in (5), one obtains that the linearized system is of the kind $\dot{x} = -\mathrm{A}x$, with the same A defined in (5). Since the symmetric part of A is $\frac{1}{2}(\mathrm{A} + \mathrm{A}^*) = \tilde{\mathrm{C}}(g, \mu, \alpha) > 0$, stability is immediately concluded from the Lyapunov function $V = \|\boldsymbol{\varphi}\|^2 + \|\boldsymbol{p}\|^2$. ∎

## VI. Proportional-derivative control

*Proposition 7 (Joint proportional-derivative control):* Assume that Condition 3 holds. Moreover, suppose that $\mathbb{I} = \|\mathbb{I}\| I$. Then the control law

$$\begin{cases} \boldsymbol{\tau} = \ \|\mathbb{I}\| \langle (\mathrm{S}y)(g - y) \rangle \ - k_d \|\mathbb{I}\| \langle (\mathrm{S}y)\, \dot{y} \rangle, \\ \boldsymbol{f} = \alpha m \langle (\nabla y)\, (g - y) \rangle - \alpha m k_d \langle (\nabla y)\, \dot{y} \rangle, \end{cases} \tag{8}$$

for a large enough $k_d$, guarantees local asymptotic stability of the second-order system (1) near $y = g$.

*Proof:* The point $(\boldsymbol{r}, \boldsymbol{\omega}, \boldsymbol{p}, \boldsymbol{v}) = (\mathrm{Id}, 0, 0, 0)$, i.e., $y = g$, is an equilibrium of the system. Make the change of variable $\boldsymbol{r} \mapsto \boldsymbol{\varphi}$, and consider the linearization around the equilibrium:

$$\begin{cases} \dot{\boldsymbol{\varphi}} = \boldsymbol{\omega}, \\ \dot{\boldsymbol{\omega}} = 0 + D_{\boldsymbol{\varphi}}\boldsymbol{\tau} \cdot \boldsymbol{\varphi} + D_{\boldsymbol{p}}\boldsymbol{\tau} \cdot \boldsymbol{p} + D_{\boldsymbol{\omega}}\boldsymbol{\tau} \cdot \boldsymbol{\omega} + D_{\boldsymbol{v}}\boldsymbol{\tau} \cdot \boldsymbol{v}, \\ \dot{\boldsymbol{p}} = \boldsymbol{v}, \\ \dot{\boldsymbol{v}} = 0 + D_{\boldsymbol{\varphi}}\boldsymbol{f} \cdot \boldsymbol{\varphi} + D_{\boldsymbol{p}}\boldsymbol{f} \cdot \boldsymbol{p} + D_{\boldsymbol{\omega}}\boldsymbol{f} \cdot \boldsymbol{\omega} + D_{\boldsymbol{v}}\boldsymbol{f} \cdot \boldsymbol{v}. \end{cases} \tag{9}$$

All the derivatives are computed at the equilibrium point. We already computed many of these derivatives as part of the proofs of earlier propositions. Let $w = [\, \boldsymbol{\varphi}\ \boldsymbol{p}\,]^*$, $z = [\, \boldsymbol{\omega}\ \boldsymbol{v}\,]^*$. The linearized dynamics can be written as

$$\begin{cases} \dot{w} = z, \\ \dot{z} \ = -\mathrm{A}w - k_d \mathrm{A} z. \end{cases}$$

We can prove stability of this system by considering a Lyapunov candidate

$$\begin{aligned} V &= w^* \mathrm{A} w + \frac{1}{2} z^* z + k_d^{-1} w^* z \\ &= \begin{bmatrix} w \\ z \end{bmatrix}^* \begin{bmatrix} \mathrm{A} + \mathrm{A}^* & \frac{1}{2} k_d^{-1} I \\ \frac{1}{2} k_d^{-1} I & \frac{1}{2} I \end{bmatrix} \begin{bmatrix} w \\ z \end{bmatrix}. \end{aligned}$$

By Condition 3, $\mathrm{A} + \mathrm{A}^* = 2\tilde{\mathrm{C}}(g, \mu_g, \alpha) > 0$; thus $V$ is positive definite for large enough $k_d$. We also compute the derivative of $V$ as follows:

$$\begin{aligned} \dot{V} &= w^* \mathrm{A} \dot{w} + \dot{w}^* \mathrm{A} w + z^* \dot{z} + k_d^{-1} w^* \dot{z} + k_d^{-1} \dot{w}^* z \\ &= w^* \mathrm{A} z + z^* \mathrm{A} w + z^* (-\mathrm{A} w - k_d \mathrm{A} z) \\ &\qquad + k_d^{-1} w^* (-\mathrm{A} w - k_d \mathrm{A} z) + k_d^{-1} z^* z \\ &= -z^* (k_d \mathrm{A} - k_d^{-1} I) z - w^* k_d^{-1} \mathrm{A} w \\ &= -z^* (k_d \tilde{\mathrm{C}}(g, \mu_g, \alpha) - k_d^{-1} I) z - w^* k_d^{-1} \tilde{\mathrm{C}}(g, \mu_g, \alpha) w. \end{aligned}$$

If $k_d$ is large enough, then $(k_d \tilde{\mathrm{C}}(g, \mu_g, \alpha) - k_d^{-1} I) > 0$, and $\dot{V}$ is negative definite. Thus the linearized system is asymptotically stable, and the original system is asymptotically stable as well. ∎

## VII. Bootstrapping the control law

Proposition 7 says that visual pose stabilization can be realized with the particularly simple control law (8). We note the following characteristics of such computation:

- It is a simple feedforward combination of sensory data.
- It is a multilinear form of $y$, $g$, $\dot{y}$, thus can be realized on a neural substrate [13].
- The nonlinearity is local (sparse/retinotopic), followed by a wide-field linear integration.

Therefore, one could say that the control law is bio-plausible. Moreover, in [28] we commented on the similarity of parts of this computation to a Reichardt correlator; in synthesis, the computation is equivalent to wide-field integration of elementary motion detectors, which is an accepted model for capturing many aspects of the visual computation in the fruit fly brain [35].

In this section, we go one step further and argue that this control law can be "bootstrapped" with Hebbian learning. This proposes an even stronger argument for bio-plausibility: in fact, instead of having to assume that the animal encodes a large number of "weights" in its genotype, one may assume that it has evolved the computational layout via a "pre-wired" generic unsupervised learning algorithm.

To see what exactly should be learned, we rewrite once again the control law (8) using a tensorial notation as follows:

$$\begin{cases} \boldsymbol{\tau}^k = \mathrm{A}_v^{uk} \left( (g^v - y^v) - k_d \dot{y}^v \right) y_u, \\ \boldsymbol{f}^k = \mathrm{B}_v^{uk} \left( (g^v - y^v) - k_d \dot{y}^v \right) y_u. \end{cases} \tag{10}$$

Recall that, following the Einstein summation convention, we integrate over repeated indices. The index $k$, not repeated, spans the three components of force and torque. This expression can be interpreted either on the continuous visual sphere ($y^s = y(s)$), or for discrete sensors ($y^i = y(s^i)$). In the first case, the indices $u, v$ span $\mathbb{S}^2$ and we interpret $\mathrm{A}, \mathrm{B}$ as generic multilinear operators; in the second case, $1 \leq u, v \leq n$, with $n$ being the number of "pixels", and $\mathrm{A}, \mathrm{B}$ can be thought as $n \times n \times 3$ tensors: they act on two vectors in $\mathbb{R}^n$ and return a vector in $\mathbb{R}^3$.

The bootstrapping procedure able to learn the control tensors $\mathrm{A}_v^{uk}$, $\mathrm{B}_v^{uk}$ is given as Algorithm 1 in the next page.

*Proposition 8 (Bootstrapped control law):* Assume Condition 3 holds for $\alpha = \mathbb{E}\{\mu\}$ and $\tilde{g} = \mathrm{P}g$. Then the control law (10), with the tensors learned by Algorithm 1, stabilizes $y \to g$ and $(\boldsymbol{\omega}, \boldsymbol{v}) \to 0$.

We need some intermediate results before proving this.

*Definition 1 (Right-invariant environment):* Write the visual input as a function of the pose $\boldsymbol{q}$ as $y(\boldsymbol{q})$. The environment is *right-invariant* if, when averaging any function of $y(\boldsymbol{q})$ over all possible poses, the result is invariant to an arbitrary roto-translation $\boldsymbol{\delta} \in \mathrm{SE}(3)$: $\mathbb{E}_{\boldsymbol{q}}\{f(y(\boldsymbol{q}))\} = \mathbb{E}_{\boldsymbol{q}}\{f(y(\boldsymbol{q}\boldsymbol{\delta}))\}$.

*Proposition 9 (Bootstrapping results):* Assume $\mathbb{I} = \|\mathbb{I}\| I$, and let the environment be right-invariant. Then the bootstrapping procedure converges, up to omitted constant factors, to

$$\mathrm{M}_s^{uk} \to \mathrm{S}^k \mathrm{P}_s^u, \qquad \mathrm{N}_s^{uk} \to \mathbb{E}\{\mu\} \nabla^k \mathrm{P}_s^u.$$

These are *smoothed* versions (using the spatial covariance as smoothing operation) of the operators S and $\nabla$.

*Proof:* We will prove the result for M and omit the analogous proof for N. Firstly, we consider the dynamics of $\boldsymbol{\omega}(t)$ in each calibration episode, under the influence of a randomly sampled constant torque $\boldsymbol{\tau}(t) = \overline{\boldsymbol{\tau}}$. Since $\mathbb{I}$ is diagonal, the dynamics of $\boldsymbol{\omega}$ are linear and, from linear systems theory [36], $\boldsymbol{\omega}(t) = c(t)\overline{\boldsymbol{\tau}}$ for some $c(t) \geq 0$. We can now compute $\mathrm{M}_s^u$ directly from (20), by summing over all episodes, and integrating over the interval $[0, T]$:

$$\begin{aligned}
\mathrm{M}_s^u &\propto \textstyle\sum_a \int_0^T \overline{\boldsymbol{\tau}}^{(a)} \dot{y}(u,t) y^*(s,t)\, dt \\
&= \textstyle\sum_a \int_0^T \overline{\boldsymbol{\tau}}^{(a)} \boldsymbol{\omega}(t)^* (\mathrm{S} y(u,t)) y^*(s,t) dt && (11)\\
&= \textstyle\sum_a \int_0^T \overline{\boldsymbol{\tau}}^{(a)} c(t) \overline{\boldsymbol{\tau}}^{(a)*} (\mathrm{S} y(u,t)) y^*(s,t) dt && (12)\\
&= \mathbb{E}_{\boldsymbol{r}_0, \overline{\boldsymbol{\tau}}} \Big\{ \overline{\boldsymbol{\tau}}\, \overline{\boldsymbol{\tau}}^* \mathrm{S} \textstyle\int_0^T c(t) y(u,t) y^*(s,t) dt \Big\} && (13)\\
&= \mathbb{E}_{\boldsymbol{r}_0, \overline{\boldsymbol{\tau}}} \Big\{ \overline{\boldsymbol{\tau}}\, \overline{\boldsymbol{\tau}}^* \mathrm{S} \textstyle\int_0^T c(t) m(\boldsymbol{r}(t)u) m^*(\boldsymbol{r}(t)s) dt \Big\} && (14)\\
&= \mathbb{E}_{\overline{\boldsymbol{\tau}}} \Big\{ \overline{\boldsymbol{\tau}}\, \overline{\boldsymbol{\tau}}^* \mathrm{S} \textstyle\int_0^T c(t) \mathbb{E}_{\boldsymbol{r}_0} \{ m(\boldsymbol{r}_0 \delta(t) u)) m^*(\boldsymbol{r}_0 \delta(t) s) \} dt \Big\} && (15)\\
&= \mathbb{E}_{\overline{\boldsymbol{\tau}}} \Big\{ \overline{\boldsymbol{\tau}}\, \overline{\boldsymbol{\tau}}^* \mathrm{S} \textstyle\int_0^T c(t) \mathbb{E}_{\boldsymbol{r}_0} \{ y(u,0) y^*(s,0) \}\, dt \Big\} && (16)\\
&\propto \mathrm{S} \mathbb{E}_{\boldsymbol{r}_0} \{ y(u,0) y^*(s,0) \} && (17)\\
&= \mathrm{S} \left( \mathrm{P}_s^u + \overline{y}^2 \mathbf{1}\mathbf{1}^* \right) = \mathrm{S}\mathrm{P}_s^u. && (18)
\end{aligned}$$

In (11), we used the expression for $\dot{y}$ in Lemma 1. In (12), we used $\boldsymbol{\omega}(t) = c(t)\overline{\boldsymbol{\tau}}$. In (13), we substituted the sum over all episodes by the expectation, assuming that we had an extensive set of samples. The expectation is taken over the generated torque $\overline{\boldsymbol{\tau}}$ and the initial attitude $\boldsymbol{r}_0$. In (14), we used the definition $y(s) = m(\boldsymbol{r}(t)s)$. In (15), we decomposed $\boldsymbol{r}(t)$ as $\boldsymbol{r}_0 \delta(t)$, where $\delta(t)$ is the time-dependent rotation induced by the applied torque. In (16), we used the definition of right-invariance: we can get rid of the initial displacement when taking the expectation with respect to all poses $\boldsymbol{r}_0$. In (17), we computed and absorbed the constants $\mathbb{E}\{\overline{\boldsymbol{\tau}}\, \overline{\boldsymbol{\tau}}^*\} = \sigma^2 I$ and $\int_0^T c(t) dt$. In (18), we used a property of the covariance, which in 1D can be written as $\mathbb{E}\{x^2\} = \mathrm{cov}(x) + \mathbb{E}\{x\}^2$. The value $\overline{y}$ is the average of $y(s)$ over all $s$, all configurations, and $\mathbf{1}$ is the constant function on the sphere. Then we used the fact that S as a differential operator annihilates the constant $\mathbf{1}\mathbf{1}^*$. Analogously, one can prove that N tends to $\mathbb{E}\{\mu\} \nabla \mathrm{P}$. ■

*Proof of Proposition 8:* Substitute in (10) the tensors A, B created by the bootstrapping procedure:

$$\begin{cases} \boldsymbol{\tau}^k = \mathrm{P}_v^s \left( \mathrm{S}^k \mathrm{P}_s^u \right) \left( (g^v - y^v) - k_d \dot{y}^v \right) y_u, \\ \boldsymbol{f}^k = \mathrm{P}_v^s \left( \overline{\mu} \nabla^k \mathrm{P}_s^u \right) \left( (g^v - y^v) - k_d \dot{y}^v \right) y_u. \end{cases} \tag{19}$$

By defining $\tilde{y}^s = \mathrm{P}_v^s y^v$, $\tilde{g}^s = \mathrm{P}_v^s g^v$, and $\tilde{y}_s = (\tilde{y}^s)^*$,

$$\begin{cases} \boldsymbol{\tau}^k = \left( (\tilde{g}^s - \tilde{y}^s) - k_d \dot{\tilde{y}}^s \right) \mathrm{S}^k \tilde{y}_s, \\ \boldsymbol{f}^k = \overline{\mu} \left( (\tilde{g}^s - \tilde{y}^s) - k_d \dot{\tilde{y}}^s \right) \nabla^k \tilde{y}_s. \end{cases}$$

Converted back to our old notation, this is equivalent to (8) under the change $g \mapsto \tilde{g}$ and $y \mapsto \tilde{y}$ (up to a constant factor, which can be incorporated in A or B). Thus we have convergence on the same conditions of Proposition 7. ■

### A. Speeding up the bootstrapping procedure

We mention two shortcuts that allow us to speed up the boostrapping procedure.

*Remark 2:* If one can sense velocities directly, then the learning rules (20) and (21) can be changed to

$$\dot{\mathrm{M}}_s^{uk} = \boldsymbol{\omega}^k \dot{y}^u y_s, \qquad \dot{\mathrm{N}}_s^{uk} = \boldsymbol{v}^k \dot{y}^u y_s.$$

The learning converges to the same values, as can be seen from adapting (actually shortening) the proof of Proposition 9.

*Remark 3:* If one knows the directions $s^i$ of each pixel, then it suffices to learn the tensor M, and recover N using $\mathrm{N} = s \times \mathrm{M}$, since $s \times \mathrm{M} = s \times (\mathrm{SP}) = s \times (s \times \nabla \mathrm{P}) = \nabla \mathrm{P}$. This method, however, while convenient for speeding up the simulation, violates our bio-plausibility criterion because it requires knowing each eye direction $s^i$, which is not bootstrappable from sensory input.

### B. What "learnable" means for an engineer

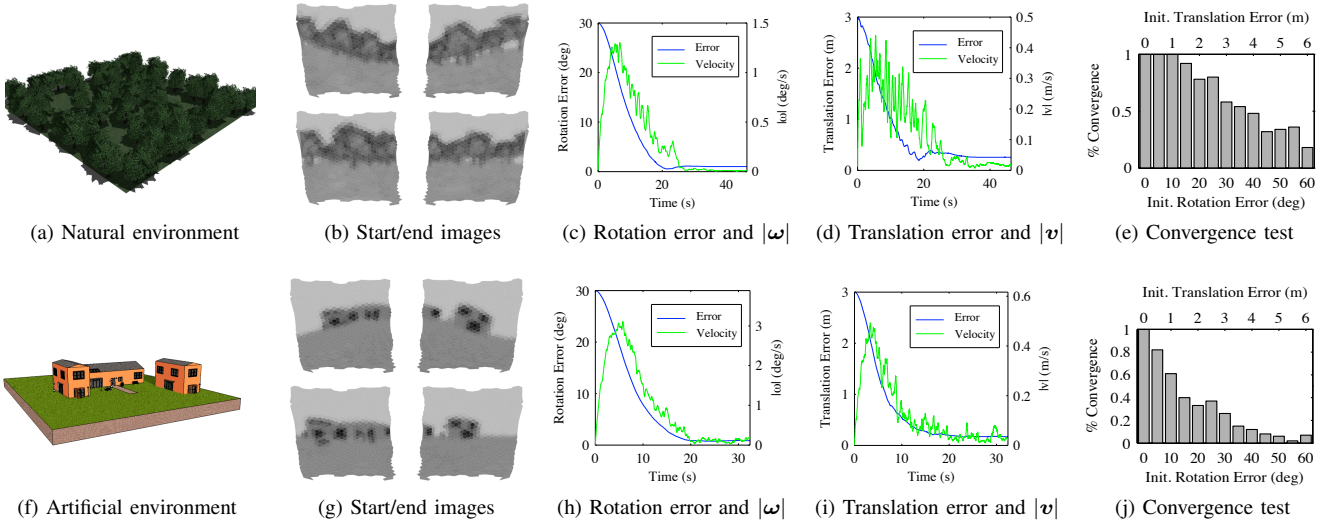
The significance of the existence of such a bootstrappable control law is clear from the biological point of view. Is it interesting also for an engineer? One interesting advantage is that this bootstrapping algorithm can be run with a *completely uncalibrated camera.* In fact, the bootstrapping procedure works with the raw values of the pixel luminance: nowhere do we need to compute anything regarding to the camera calibration parameters. For example, we can use a camera setup with a completely irregular mirror surface, without worrying about a complex camera calibration procedure.

**Algorithm 1** Bootstrapping procedure

**Assumptions:** The system has naturally damped dynamics with $\epsilon_{\boldsymbol{v}}, \epsilon_{\boldsymbol{\omega}} > 0$; the environment is right-invariant.

**Output:** Control tensors $\mathrm{A}_v^{uk}$, $\mathrm{B}_v^{uk}$.

1) Initialize M and N: $\mathrm{M}_s^{uk} = 0$, $\mathrm{N}_s^{uk} = 0$.
2) For learning M, repeat a large number of times:
   - Start from rest ($\boldsymbol{\omega} = 0$, $\boldsymbol{v} = 0$) with a random attitude and position $(\boldsymbol{r}, \boldsymbol{p})$.
   - Sample a test torque $\overline{\boldsymbol{\tau}}$ from some distribution such that $\mathbb{E}\{\overline{\boldsymbol{\tau}}\} = 0$ and $\mathbb{E}\{\overline{\boldsymbol{\tau}}\, \overline{\boldsymbol{\tau}}^*\} = \sigma^2 I$ .
   - Apply $\overline{\boldsymbol{\tau}}$ for a duration of $T$; evolve the tensor $\mathrm{M}_s^{uk}$ using
   $$\dot{\mathrm{M}}_s^{uk} = \overline{\boldsymbol{\tau}}^k \dot{y}^u y_s. \tag{20}$$
   (we omit a decaying term to keep M bounded)
3) Analogously, do the same for the tensor N: this time sample the force $\overline{\boldsymbol{f}}$, and evolve using
   $$\dot{\mathrm{N}}_s^{uk} = \overline{\boldsymbol{f}}^k \dot{y}^u y_s. \tag{21}$$
4) Using any known Hebbian algorithm, learn the covariance $\mathrm{P}_s^u \triangleq \mathrm{cov}(y^u, y_s)$.
5) Let $\mathrm{A}_v^{uk} = \mathrm{P}_v^s \mathrm{M}_s^{uk}$ and $\mathrm{B}_v^{uk} = \mathrm{P}_v^s \mathrm{N}_s^{uk}$.




(a) Natural environment (b) Start/end images (c) Rotation error and $|\boldsymbol{\omega}|$ (d) Translation error and $|\boldsymbol{v}|$ (e) Convergence test

(f) Artificial environment (g) Start/end images (h) Rotation error and $|\boldsymbol{\omega}|$ (i) Translation error and $|\boldsymbol{v}|$ (j) Convergence test

Figure 3. Simulation results. We visualize the rotation error using the geodesic distance on SO(3) [30]. In all simulations, we determined the length scale to ensure that the features have reasonable realistic size (e.g. the height of a tree is 10–15 m), and the time scale to obtain physically achievable kinematics (e.g. the fruit fly has a typical translational velocity of 0.2 m/s and maximum of 1.2 m/s)

## VIII. Simulations

We used *fsee* [37] to simulate the visual input of the fruit fly *Drosophila* with a biologically realistic model, which includes the spatial disposition of the 1,398 "pixels" that form the compound eye, the spatial blurring in the optics, and temporal filtering of the underlying photoreceptors, according to current knowledge in visual physiology [37]. Fig. 1, 3b, 3g show typical simulations of the visual stimulus. The 1,398 ommatidia of a fly cover almost all the visual sphere; in the images, they are arranged using a cylindrical projection. In the simulations, we approximate $\dot{y}(t)$ as $\frac{1}{\Delta t}[\,y(t) - y(t-\Delta t)]$.

We used two environment models: a natural environment (Fig. 3a) and an artificial one (Fig. 3f) [38]. 20,000 episodes were used to learn the tensors according to the bootstrapping procedure. We used the shortcuts mentioned in Remarks 2–3 to speed up the learning: the bootstrapping of N is intrinsically difficult for our simulation setup, because providing exhaustive sampling of the nearness profile $\mu(s)$ requires an elaborate collection of 3D models; this seems very inconvenient unless the simulation environment is equipped with automated generation of such a collection.

A typical convergent example using the proportional-derivative control law (8) is also shown for each environment as in Fig. 3c, 3h (rotation component) and Fig. 3d, 3i (translation component). The rotation error settles to about 1 deg, which is reasonable given the discretization of the sensor: each "pixel" covers about 5 steradians. This nonzero rotation error induces a corresponding small nonzero translation error. The nonzero, in fact oscillating, final velocities are probably due to either the temporal discretization of $\dot{y}$ or the quantization (8-bit) of the simulated luminance profile.

Since the control law is only guaranteed to be locally stable, we also used the Monte Carlo method to perform numerical tests of the region of attraction. The results are reported in Fig. 3e and 3j for the two environments, for increasing, randomly sampled initial rotation and translation error. In both environments, the control law provides a reasonable region of attraction. Despite the forest environment having less distinctive features than the house environment, it shows a larger region of attraction, presumably because the omnidirectional nonzero $\mu_g(s)$ provides better localization. Another possible reason is that the other facades of the house can be observed if the initial pose deviates too much from the goal, which leads to a violation of the non-occlusion assumption.

## IX. Conclusions

We showed that the task of visual servoing can be solved (at least locally) through purely visual control laws. The non-obvious result is that, even though the Jacobian of the error depends on the nearness, this can be ignored in the control law while still attaining local stability. These control laws respect the constraints of bio-plausible computation by using a bilinear/quadratic operation on the raw luminance: in a traditional implementation, the operation is only a tensor-vector multiplication with the raw pixel array. Moreover, these laws can be bootstrapped by a learning agent. In the biological domain, this means that the genotype may encode only an adaptive rule, while from the engineering viewpoint, it means that, in principle, an adaptive zero-configuration system can be realized.

We are in the process of extending our work in two directions. From the engineering viewpoint, we want to ascertain whether the approach can give solid advantages over traditional methods (feature-based and not). This would imply looking at issues such as change of lighting condition, occlusions, locality, nonholonomic constraints, which we did not consider yet. At a more abstract level, we want to investigate whether the same bio-plausible/bootstrappable approach is

applicable for more complex tasks, such as structure from motion and obstacle avoidance.

**Acknowledgements**. This work is supported in part by the US Army Institute for Collaborative Biotechnology (ICB) and the Boeing Corporation.

## APPENDIX

We group here some technical lemmas. These are not new results and appear, often in a disguised form with different notation, in many other papers. The formulas are valid with the assumptions that: 1) $y$ is differentiable; and 2) the nearness profile $\mu(s)$ is continuous (i.e., there are no occlusions).

*Lemma 1:* $\dot{y}(s,t) = \mu(s)\nabla_s y(s,t)^* \boldsymbol{v} + (s \times \nabla_s y(s,t))^* \boldsymbol{\omega}$, or more compactly, $\dot{y} = \mu \nabla y^* \boldsymbol{v} + \mathrm{S}y^* \boldsymbol{\omega}$.

*Proof:* Use $\dot{s} = \mu(s)(I - ss^*)v + \omega \times s$ [39] and $\nabla_s y \cdot (I - ss^*) = \nabla_s y$. The lemma follows from the chain rule. ■

*Lemma 2:* $D_p y \cdot v = \mu(\nabla y)^*$ and $D_{\boldsymbol{r}} y \cdot \boldsymbol{\omega}^\wedge = (\mathrm{S}y)^* \boldsymbol{\omega}$.

*Proof:* Note $\dot{y} = D_p y \cdot \dot{p} + D_r y \cdot \dot{r}$ and use Lemma 1. ■

*Lemma 3:* $D_{\boldsymbol{\varphi}} y|_{\boldsymbol{\varphi}=0} \cdot u = (\mathrm{S}y)^* u$.

*Proof:* For a generic function of $\boldsymbol{r}$, we have that $D_{\boldsymbol{\varphi}}(f(\boldsymbol{r}))|_{\boldsymbol{\varphi}=0} \cdot u = D_{\boldsymbol{r}}(f(\boldsymbol{r}))|_{\boldsymbol{r}=\mathrm{Id}} \circ D_{\boldsymbol{\varphi}} \mathrm{Exp}(\boldsymbol{\varphi}^\wedge)|_{\boldsymbol{\varphi}=0} \cdot u = D_{\boldsymbol{r}}(f(\boldsymbol{r}))|_{\boldsymbol{r}=\mathrm{Id}} \cdot u^\wedge$. In particular, for $f = y$, we have that $D_{\boldsymbol{r}} y(s,t)|_{\boldsymbol{r}=\mathrm{Id}} \cdot \Omega = D_{\boldsymbol{r}} m(\boldsymbol{r}(t)s)|_{\boldsymbol{r}=\mathrm{Id}} \cdot \Omega = \nabla_s y(s)^* \Omega s = (s \times \nabla_s y(s)^* \Omega^\vee$. Thus $D_{\boldsymbol{\varphi}} y(s,t)|_{\boldsymbol{\varphi}=0} \cdot u = D_{\boldsymbol{r}} y|_{\boldsymbol{r}=\mathrm{Id}} \cdot u^\wedge = (s \times \nabla_s y(s)^* u \triangleq (\mathrm{S}y)^* u$. ■

## REFERENCES


[1] W. Reichardt and T. Poggio, "Visual control of orientation behaviour in the fly. Part I. A quantitative analysis.," *Quarterly Reviews of Biophysics*, vol. 9, no. 3, p. 311, 1976.

[2] C. David, "Visual control of the partition of flight force between lift and thrust in free-flying *Drosophila*," *Nature*, vol. 313, pp. 48 – 50, 1985.

[3] B. Cartwright and T. Collett, "How honey bees use landmarks to guide their return to a food source," *Nature*, vol. 295, no. 5850, 1982.

[4] T. S. Collett and M. Collett, "Memory use in insect visual navigation," *Nature Reviews Neuroscience*, vol. 3, pp. 542–552, 2002.

[5] K. Cheng, T. S. Collett, A. Pickhard, and R. Wehner, "The use of visual landmarks by honeybees: Bees weight landmarks according to their distance from the goal," *J. of Comparative Physiology, A*, vol. 161, pp. 469–475, 1987.

[6] J. Zeil, "Orientation flights of solitary wasps (*Cerceris*; sphecidae; hymenoptera). I. Description of flights," *J. of Comparative Physiology A*, vol. 172, pp. 189–205, 1993.

[7] T. Collett, "Insect navigation en route to the goal: Multiple strategies for the use of landmarks.," *J. of Experimental Biology*, vol. 199, 1996.

[8] K. Fauria, R. Campan, and A. Grimal, "Visual marks learned by the solitary bee *Megachile rotundata* for localizing its nest," *Animal Behaviour*, vol. 67, pp. 523–530, 2004.

[9] R. Wehner and S. Wehner, "Insect navigation: use of maps or Ariadne's thread?," *Ethol. Ecol. Evol.*, vol. 2, pp. 27–48, 1990.

[10] T. Vladusich, J. M. Hemmi, M. V. Srinivasan, and J. Zeil, "Interactions of visual odometry and landmark guidance during food search in honeybees," *J. Exp. Biol.*, vol. 208, pp. 4123–4135, 2005.

[11] R. Menzel, U. Greggers, A. Smith, S. Berger, R. Brandt, S. Brunke, G. Bundrock, S. Hülse, T. Plümpe, F. Schaupp, E. Schüttler, S. Stach, J. Stindt, N. Stollhoff, and N. Watzl, "Honey bees navigate according to a map-like spatial memory," *Proceedings of the National Academy of Sciences USA*, vol. 102, no. 8, pp. 3040–3045, 2005.

[12] S. Fry and R. Wehner, "Look and turn: Landmark-based goal-navigation in honey bees," *J. of Experimental Biology*, vol. 208, 2005.

[13] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press, October 2004.

[14] V. Kallem, M. Dewan, J. Swensen, G. Hager, and N. Cowan, "Kernel-based visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and System*, 2007.

[15] W. Stürzl and J. Zeil, "Depth, contrast and view-based homing in outdoor scenes," *Biological Cybernetics*, vol. 96, no. 5, pp. 519–531, 2007.

[16] C. Collewet, E. Marchand, and F. Chaumette, "Visual servoing set free from image processing," in *IEEE Conf. on Robotics and Automation*, 2008.

[17] A. Dame and E. Marchand, "Entropy-based visual servoing," in *IEEE Conf. on Robotics and Automation*, 2009.

[18] J. Zeil, M. I. Hofmann, and J. S. Chahl, "Catchment areas of panoramic snapshots in outdoor scenes," *J. Opt. Soc. Am. A*, vol. 20, no. 3, 2003.

[19] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *Trans. on Robotics and Automation*, vol. 12, no. 5, 1996.

[20] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Mag.*, vol. 13, no. 4, 2006.

[21] R. Vassallo, H. Schneebeli, and J. Santos-Victor, "Visual servoing and appearance for navigation," *Robotics and Autonomous Systems*, vol. 31, no. 1, pp. 87–98, 2000.

[22] D. Fontanelli, A. Danesi, F. A. W. Belo, P. Salaris, and A. Bicchi, "Visual servoing in the large," *Int. J. of Robotics Research*, vol. 28, no. 6, pp. 802–814, 2009.

[23] É. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, 2005.

[24] K. Deguchi, "A direct interpretation of dynamic images with camera and object motions for vision guided robot control," *Int. J. of Computer Vision*, vol. 37, no. 1, pp. 7–20, 2000.

[25] J. Piepmeier, G. McMurray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *Trans. on Robotics and Automation*, 2004.

[26] A. Farahmand, A. Shademan, M. Jagersand, and C. Szepesvari, "Model-based and model-free reinforcement learning for visual servoing," in *IEEE Conf. on Robotics and Automation*, 2009.

[27] B. Kuipers, *Robot and Cognitive Approaches to Spatial Mapping.*, ch. An intellectual history of the Spatial Semantic Hierarchy. Springer Verlag, 2008.

[28] A. Censi, S. Han, S. B. Fuller, and R. M. Murray, "A bio-plausible design for visual attitude stabilization," in *IEEE Conf. on Decision and Control*, 2009.

[29] R. Abraham, J. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications*. Springer, 1988.

[30] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[31] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, 2008.

[32] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun, "Structure from motion without correspondence," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[33] A. J. Yezzi and S. Soatto, "Structure from motion for scenes without features," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 525–532, June 2003.

[34] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature Depth Observation for Image-based Visual Servoing: Theory and Experiments," *Int. J. of Robotics Research*, vol. 27, no. 10, pp. 1093–1116, 2008.

[35] J. Lindemann, H. Weiss, R. Möller, and M. Egelhaaf, "Saccadic flight strategy facilitates collision avoidance: closed-loop performance of a cyberfly," *Biological Cybernetics*, vol. 98, no. 3, pp. 213–227, 2008.

[36] B. Friedland, *Control Systems Design: An Introduction to State-Space Methods*. McGraw-Hill, 1985.

[37] W. B. Dickson, A. D. Straw, and M. H. Dickinson, "Integrative model of *Drosophila* flight," *AIAA Journal*, vol. 46, pp. 2150–2164, 2008.

[38] Google 3D Warehouse ID: 47e2c8beeae865551282260810482380 (house) and ID: 6b4912bc2c4b2df85e28caa3b26a73fd (forest).

[39] J. Koenderink and A. Doorn, "Facts on optic flow," *Biological Cybernetics*, vol. 56, no. 4, pp. 247–254, 1987.