HAL open science

Bayesian Action-Perception loop modeling: Application to trajectory generation and recognition using internal motor simulation

Estelle Gilet, Julien Diard, Richard Palluel-Germain, Pierre Bessière

To cite this version:

Estelle Gilet, Julien Diard, Richard Palluel-Germain, Pierre Bessière. Bayesian Action-Perception loop modeling: Application to trajectory generation and recognition using internal motor simulation. Thirtieth International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (Maxent 2010), Jul 2010, Chamonix, France. pp.59–66. hal-00961128

HAL Id: hal-00961128

https://hal.science/hal-00961128v1

Submitted on 19 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Bayesian Action-Perception loop modeling: Application to trajectory generation and recognition using internal motor simulation

Estelle Gilet*, Julien Diard[1,†], Richard Palluel-Germain† and Pierre Bessière*

*Laboratoire d'Informatique de Grenoble - INRIA Rhône-Alpes - CNRS
655 av. de l'Europe, 38330 Montbonnot, France
†Laboratoire de Psychologie et NeuroCognition - CNRS
Université Pierre-Mendès-France, BSHM, BP 47, 38040 Grenoble, France

Abstract. This paper is about modeling perception-action loops and, more precisely, the study of the influence of motor knowledge during perception tasks. We use the Bayesian Action-Perception (BAP) model, which deals with the sensorimotor loop involved in reading and writing cursive isolated letters and includes an internal simulation of movement loop. By using this probabilistic model we simulate letter recognition, both with and without internal motor simulation. Comparison of their performance yields an experimental prediction, which we set forth.

Keywords: Sensorimotor loop, Bayesian modeling, Motor simulation
PACS: 87.85.Ng

INTRODUCTION

Some people enjoy watching sport events, while others do not. The reason, the "why", is probably irrelevant to scientific curiosity; tastes differ, personalities and temperaments also. However, the manner, the "how" of these differences, is worthy of attention. Indeed, the general principles underlying the numerous and complicated processes involved in perception and (in this case) in perception of actions, are yet to be understood.

Recent observations suggest that perception of performed actions is based not only on sensory cues, but also on internal simulation of actions [1, 2], especially if the action performed is part of the perceiving subject's action repertoire [3].

Instead of sport actions, which usually involve complicated articulated systems, we study here the perception-action loop involved in handwriting and reading. In the case of this simpler system, too, one finds evidence that the motor system is involved in letter recognition. Behavioral [4, 5, 6, 7] and neuro-imaging studies [8, 9] both support this idea.

For instance, Longcamp et al. have studied the activation of motor areas of the brain during writing and reading tasks [8]. They observed that a part of the motor cortex is significantly activated during both tasks. This is surprising for the reading task: although the subjects remain still, a motor area was activated. Another class of stimuli was

[1] To whom correspondence should be addressed.

presented: pseudo-letters, which are visually as complex as letters, but for which the subjects have no previous experience in writing. When such pseudo-letters were visually presented, the same motor area was not activated.

A much discussed interpretation of these observations is that perceiving a letter would entail a motor simulation of movements associated with the writing of that letter. This mechanism would, it is supposed, improve perceptual recognition.

The Bayesian Action-Perception (BAP) model is a model of the entire perception action loop involved in reading and writing cursive, isolated letters [10]. It is composed of perception, action and internal representation components, and also includes a model of internal simulation of movements, in the form of a feedback loop. This loop starts from the letter representation, goes to generated trajectories and back to the letter representation: it it both simulated action and simulated perception.

In this paper, we first show how this proposed mechanism for internal simulation is defined mathematically. We then use it to compare perception processes with and without internal simulation of movements. The analysis of recognition performance indicates that, in easy situations, the internal simulation does not improve performance, whereas, in difficult situations, it helps in retrieving some of the missing information. This observation is the basis for an experimental prediction, which we detail.

BAP MODEL: ASSUMPTIONS AND MODEL ARCHITECTURE

We briefly here summarize the main components of the BAP model. Its global architecture is shown Fig. 1.

The model is articulated around internal representations C_{LV} and C_{LP} of letters, one for each considered letter L and writer W. These act as a pivot between the perception model and the action model.

Letter representation models are based on two main hypotheses. First, letters are encoded in the Cartesian reference frame (the workspace). This is an effector-independent representation and ensures motor equivalence, which is the fact that writing can be performed with any effector without preliminary learning and with recognizable resulting writing style [11]. Second, letters are not encoded as complete trajectories, but are summarized as sequences of via-points along the trajectory. These via-points are placed at the starting position, the ending position, at cusps and points of the trajectory where either the vertical or horizontal velocity is zero (in other words, where the tangent is either vertical or horizontal). At each of these via-points, probability distributions about the X, Y positions and velocities are memorized.

The perception model relates the input (visual) trajectory V to the letter representation model, using a probabilistic term $P(C_{LV} \mid V)$ that describes how via-points are extracted. The action model is composed of two parts. The first is a trajectory generation phase, which relates the letter representation to a complete prepared trajectory. It is centered on a $P(P \mid C_{LP})$ term, which uses the acceleration-minimization algorithm to provide intermediary trajectories between given via-points. The second part of the action model is an effector model $P(E \mid P)$, which describes the geometric and kinematic models of the effector, in order to translate the generated trajectory P to effector commands.

The BAP model is translated, in the Bayesian Programming framework, into a joint

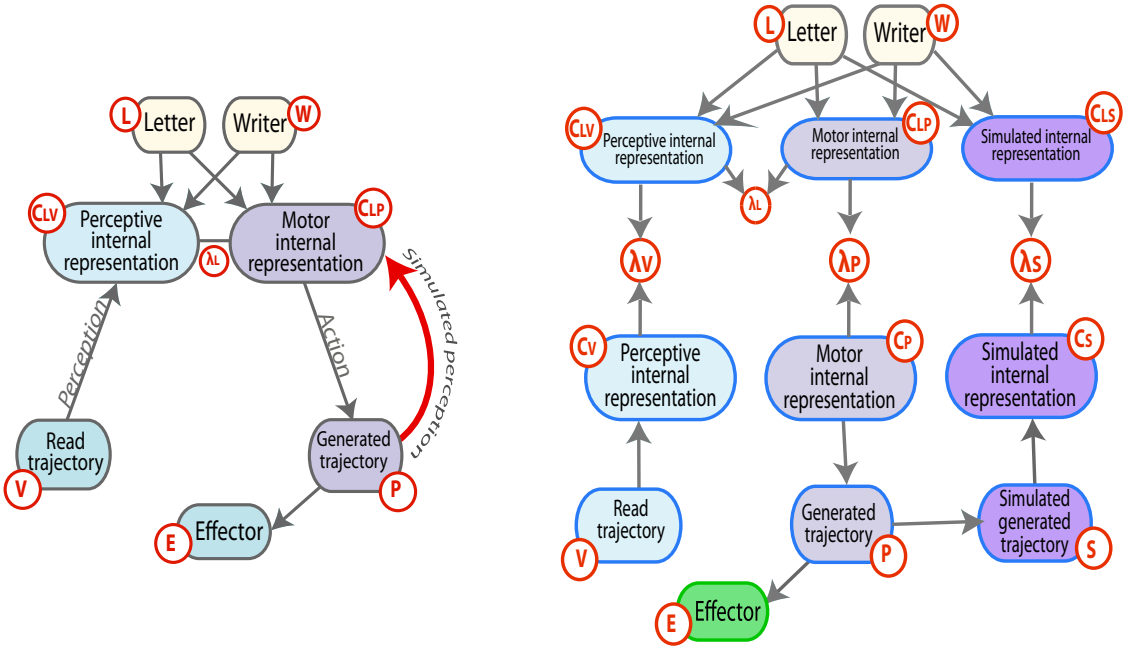

FIGURE 1. Left: architecture of the BAP model. Right: corresponding acyclic graph for defining the joint decomposition, including λ variables used as probabilistic switches.

probability distribution. However, the schema of Fig. 1 cannot be directly encoded as a probabilistic dependency structure. Indeed, the feedback loop does not correspond to any valid application of Bayes' rule. We circumvent this problem with duplication of nodes: the motor model is coupled with a simulated perception model. In other words, the generated trajectory is copied, with $P(S \mid P)$, as an input to a simulated perception model $P(C_S \mid S)$.

The other technicality is that we include, in the model, λ coherence variables [12]. These serve as probabilistic switches, allowing us to deactivate sub-models explicitly during inference. For instance, this is used to compare perception with internal simulation (the whole model is activated) with perception without internal simulation (the action model and simulated perception are deactivated). The technical overhead is a duplication of nodes around the λ variables. For instance, C_{LP} is duplicated into C_{LP} and C_P, and the $P(P \mid C_{LP})$ term described above becomes $P(P \mid C_P)$, etc.

The resulting joint decomposition is as follows (see Fig. 1, right):

$$
\begin{aligned}
& P(L\,W\,C_{LV}\,C_V\,C_{LP}\,C_P\,C_{LS}\,C_S\,V\,P\,E\,\lambda_L\,\lambda_V\,\lambda_P\,\lambda_S) \\
= \;& P(L)P(W) \\
& P(C_{LV} \mid L\,W)P(C_{LP} \mid L\,W)P(C_{LS} \mid L\,W) \\
& P(C_V \mid V)P(V)P(P \mid C_P)P(C_P)P(E \mid P)P(S \mid P)P(C_S \mid S) \\
& P(\lambda_V \mid C_{LV}\,C_V)P(\lambda_P \mid C_{LP}\,C_P)P(\lambda_S \mid C_{LS}\,C_S)P(\lambda_L \mid C_{LV}\,C_{LP})\;.
\end{aligned}
$$

Each term of the form $P(C_L \mid L\,W)$ is itself structured, because a sequence of N via-points is actually represented using a series of N variables $C_L = C_L^{0:N}$. Furthermore, each

C_L^n is itself the conjunction of position C_{Lx}^n, C_{Ly}^n variables and velocity $C_{L\dot{x}}^n, C_{L\dot{y}}^n$ variables. This yields the following decomposition:

$$
\begin{aligned}
& P(C_L^{0:N} \mid L\,W) \\
& \quad = \; P(C_{Lx}^0 \mid L\,W) P(C_{Ly}^0 \mid L\,W) P(C_{L\dot{x}}^0 \mid L\,W) P(C_{L\dot{y}}^0 \mid L\,W) \\
& \qquad \prod_{n=1}^{N} P(C_{Lx}^n \mid C_{Lx}^{n-1}\; L\,W) P(C_{Ly}^n \mid C_{Ly}^{n-1}\; L\,W) P(C_{L\dot{x}}^n \mid C_{L\dot{x}}^{n-1}\; L\,W) P(C_{L\dot{y}}^n \mid C_{L\dot{y}}^{n-1}\; L\,W) \; .
\end{aligned}
$$

For full mathematical definitions of the terms in this model, the reader should refer to Gilet's PhD thesis [10].

BAP MODEL SIMULATION EXPERIMENTS

We now present experimental results using the BAP model: we simulate recognition tasks both with and without internal simulation of movements.

Experimental environment and parameter identification

We first set the parameters of the internal representation of letters, by a data collection procedure and learning phase. Using a Wacom Intuos 3 pen tablet, we asked 4 adults to write 40 sample trajectories for each of 22 letters, providing a complete database of 3,520 trajectories. We considered only letters without pen-up: we removed *i*s, *j*s, *t*s and *x*s (as in [13]).

For each letter L, writer W and via-point position $C_{Lx}^{n-1}, C_{Ly}^{n-1}$ and velocity $C_{L\dot{x}}^{n-1}, C_{L\dot{y}}^{n-1}$, we obtain the number of observations p_i of each via-point position and velocity (at index N). These are then used to compute the parameters of Laplace succession laws for the terms of the form $P(C_{Lx}^n \mid C_{Lx}^{n-1}\; L\,W)$ in letter representation models.

With all the parameters of the model defined, Bayesian inference is used to automatically compute any probabilistic term of interest. A general-purpose probabilistic engine (ProBT© of ProBayes) is used for all of the following inferences.

Reading letters without internal simulation of movement

The cognitive task of letter recognition comprises identifying an input trajectory. In other words, the question is: "given a trajectory produced by a known writer, what is the letter?" Based on Bayesian inference in the BAP model, with only the perception and letter representation models activated, this amounts to:

$$
\begin{aligned}
& P(L \mid [V_x^{0:M} = v_x^{0:M}][V_y^{0:M} = v_y^{0:M}][W = w][\lambda_V = 1]) \\
& \quad \propto \; P([C_{LV} = f(v_x^{0:M}, v_y^{0:M})] \mid L\,[W = w]) \; ,
\end{aligned}
$$

where $V_x^{0:M}, V_y^{0:M}$ is the input trajectory, w is the specified writer and f is the via-point extraction function.

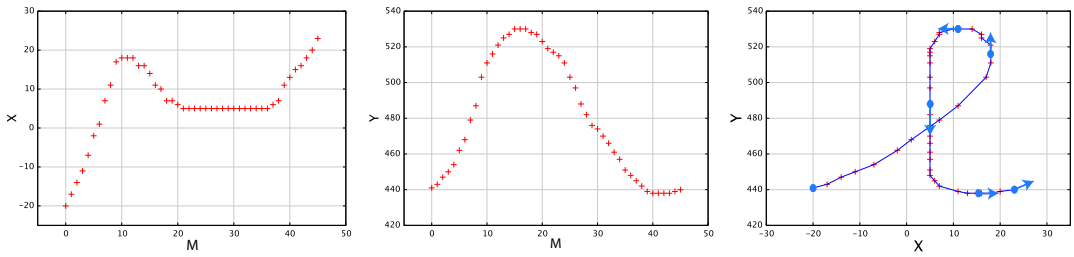

FIGURE 2. X and Y position profiles (left and center), and the resulting input trajectory (an l), with extracted via-points shown as vectors (right).

This probabilistic equation can be explained by an algorithmic equivalent. Indeed, the computation proceeds as if the via-points extracted from the input trajectory were matched to the representations learned. For each via-point and each possible letter, both positions and velocities are compared, using the memorized probability distributions: "if the letter was an a, what would be the probabilities of observing the positions and velocities of the first observed via-point?", etc.

When presented with input trajectories, the model computes the probability distribution over letters; it is usually very close to a delta (Dirac) probability distribution. Fig. 2 shows an example where the model correctly classifies the input trajectory as an l.

We ran a systematic experiment over our database of sample trajectories: out of our 40 trajectories, 35 were used for learning parameters and 5 were used for testing. We repeated this procedure using a classical K-fold cross-validation method. The result was a complete confusion matrix (which we do not detail here for lack of space). However, aggregating the correct classifications yields a performance indicator: overall, we obtained a satisfying correct recognition rate of 93.4%. Misclassifications arose due to the geometric similitude of some letters.

Reading letters with internal simulation of movements

We reproduced the same experiment as above in a different setting: we allowed the entire BAP model to function during the perception task. This is done by setting all λ variables to one and computing:

$$\begin{aligned} & P(L \mid [V_x^{0:M} = v_x^{0:M}][V_y^{0:M} = v_y^{0:M}][W = w][\lambda_V = 1][\lambda_L = 1][\lambda_P = 1][\lambda_S = 1]) \\ \propto \quad & P([C_{LV} = f(v_x^{0:M}, v_y^{0:M})] \mid L\,[W = w]) P([C_{LS} = h(g(C_{LV}))] \mid L\,[W = w]) \;. \end{aligned}$$

The result of this inference is the product of two terms: the first term is the same as previously, when the internal simulation of movements was deactivated. The second term is the result of the simulation loop: given the via-points extracted from the input trajectory, a trajectory is generated internally by the action model (function g), which is then analyzed by simulated perception (function h).

In the same setting as before, with our 40-trajectory database, we obtain a correct recognition rate of 90.2%. An analysis of the confusion matrices of both experiments (not shown) indicates that specific errors differ: some letters which were misclassified in

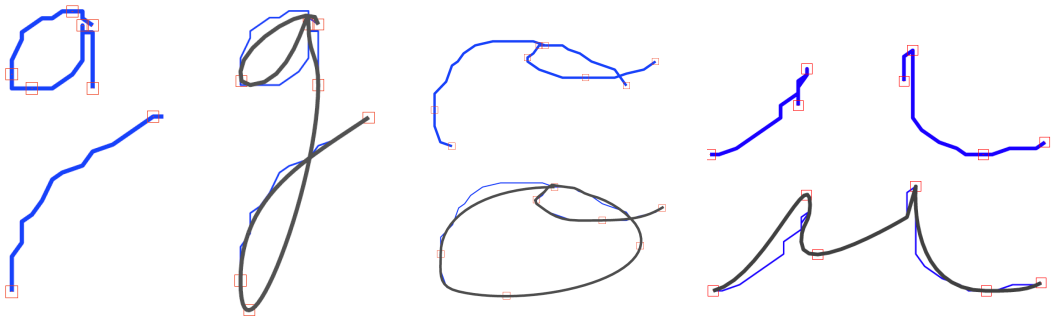

FIGURE 3. Examples of incomplete trajectories presented to the perception algorithm, and extracted via-points (boxes). The motor simulation produces a complete trajectory (smooth trajectories superposed to the initial, incomplete trajectories), which is then analyzed by simulated perception.

the reading task without simulated perception were recognized correctly using simulated perception, and vice versa. Overall, the misclassification rates are of the same magnitude in both conditions (90 vs. 93%).

Reading truncated letters

We also designed another experiment with a more difficult scenario. Instead of presenting complete trajectories as input, we designed truncated versions of trajectories, that is to say, trajectories where we erased a set of consecutive points.

In this context, we found several cases where reading without motor simulation would fail to recognize the input trajectory, whereas reading with motor simulation would succeed. We show in Fig. 3 some examples of truncated letters.

Consider for instance the letter *g* shown on the left of Fig. 3. Without internal simulation, the perception algorithm incorrectly classifies the input trajectory as a *q*. However, when internal simulation is activated, the motor model "fills the gap" by generating a complete trajectory joining the via-points; the resulting complete trajectory is analyzed back by simulated perception. This added information causes the perception algorithm to recognize the letter correctly as a *g*.

EXPERIMENTAL PREDICTION

We summarize our observations as follows: in the BAP model, in the easy case of complete letters, internal simulation of movements does not improve performance. In the harder case of truncated letters, however, internal simulation helps to correct misclassifications.

This is a hypothesis that we now pursue experimentally, in two directions. First, it would be fruitful to make this prediction more precise by quantifying the difference in performance in the case of truncated letters. This would require a more complete (or a reference) learning database, which is as yet unavailable.

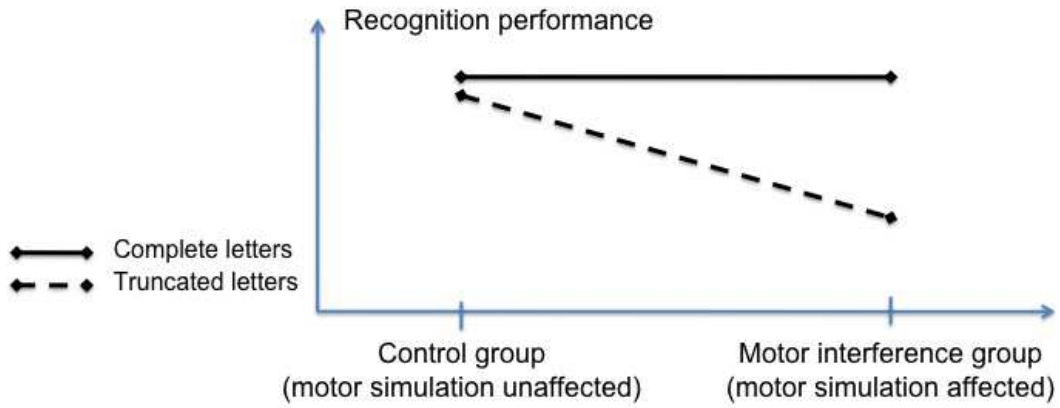

FIGURE 4. Experimental prediction of recognition performance as a function of stimulus difficulty (complete vs. truncated letters) and use of motor simulation (control vs. concurrent motor task).

Second, we have designed a preliminary psychology experiment in which participants have to recognize letters in various difficulty settings. The protocol is as follows. Letters are shown to the participant in a manner similar to the way trajectories are given to the model: since the model is given both sequence and velocity information, the participants see a moving point that follows the trajectory (contrary to just seeing the resulting trace, which would be a more natural task). The trajectory is either a complete letter or a truncated letter. In order to minimize the impact of particularities of the presented trajectories, we asked an elementary school teacher to provide reference trajectories to be used as stimuli (assuming that letters used to teach children are canonical in some way).

The truncated versions of trajectories have been designed by removing 25% of their points, either in the first or the second half of the trajectory, with a random starting position for deletion (protecting the first and last 5%, however). Deletions are treated as pen-ups: instead of showing a white dot on a black background, the dot turns black for the truncated portion, so that it is invisible. This guarantees that temporal properties of the trajectories are not affected.

The task is letter recognition, under two conditions: a control condition, and a condition in which a concurrent motor task is performed. Indeed, it has been demonstrated that a simple task like squeezing a ball tightly would be sufficient to affect, and possibly prevent, the use of internal motor simulation [14]. It is also known that motor interference influences letter recognition [7]. The expected observation is an interaction effect between condition and difficulty of the stimulus, as shown in Fig. 4.

CONCLUSION

We have presented some features of the BAP model, which is a model of the perception-action loop involved in recognition and production of handwritten characters. More precisely, we have focused on the internal simulation of movement loop. This is used to simulate and compare two cognitive tasks of letter recognition, depending on whether

this loop is activated or not. Experimental results show that internal motor simulation does not improve performance for easy stimuli, whereas it does for difficult stimuli such as truncated trajectories. This finding generates a prediction, which is the basis of a psychology experimental protocol that is currently underway.

In conclusion, we wish to highlight another issue that could benefit from our modeling of the internal simulation of movements. This concerns what is called "online" and "offline" character recognition. Online character recognition is the situation we have treated so far: the input includes velocity and sequence information. This corresponds to tasks in which the observer sees the letter as it is being traced. It is arguably less natural than offline recognition, where the input is the resulting static trace.

Offline recognition can be tackled in two ways. On the one hand, it can be treated as a problem of a different nature than online recognition, with approaches such as common optical character recognition methods (OCR). On the other hand, offline recognition can be reduced to online recognition, after a first step that finds the time sequence of the trajectory, given the input static image. To illustrate this, consider the simpler problem of finding the starting point of the trajectory, given a sequence of pixels. We believe that motor knowledge could play a crucial role, by ruling out most points that would be perceptual candidates, but would not be associated with typical motor programs.

ACKNOWLEDGMENTS

We thank J.-P. Orliaguet, E. Gentaz, J.-L. Schwartz, G. Baud-Bovy, J. Droulez and P. Baraduc for countless helpful discussions. This work has been supported by the BACS European project (FP6-IST-027140).

REFERENCES

1. M. Jeannerod, *NeuroImage* **14**, 103–109 (2001).
2. A. Berthoz, *The Brain's Sense of Movement*, Harvard University Press, Cambridge, MA, 2000.
3. B. Calvo-Merino, D. Glaser, J. Grèzes, R. Passingham, and P. Haggard, *Cerebral Cortex* **15**, 1243–1249 (2004).
4. J.-P. Orliaguet, S. Kandel, and L.-J. Boë, *Perception* **26**, 905–912 (1997).
5. G. Knoblich, E. Seigerschmidt, R. Flach, and W. Prinz, *The Quarterly Journal of Experimental Psychology* **55A**, 1027–1046 (2002).
6. J.-L. Li, and S.-L. Yeh, *Visual Cognition* **10**, 537–547 (2003).
7. K. H. James, and I. Gauthier, *Journal of Experimental Psychology: General* **138**, 416–431 (2009).
8. M. Longcamp, J.-L. Anton, M. Roth, and J.-L. Velay, *NeuroImage* **19**, 1492–1500 (2003).
9. M. Longcamp, T. Tanskanen, and R. Hari, *NeuroImage* **33**, 681–688 (2006).
10. E. Gilet, *Modélisation Bayésienne d'une boucle perception-action : application à la lecture et à l'écriture*, Thèse de doctorat, Université Joseph Fourier – Grenoble, Grenoble, France (2009).
11. N. Bernstein, *The Co-ordination and Regulation of Movements*, Oxford: Pergamon Press, 1967.
12. C. Pradalier, F. Colas, and P. Bessière, "Expressing Bayesian Fusion as a Product of Distributions: Applications in Robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS03)*, 2003, vol. 2, pp. 1851–1856.
13. Y. Wada, Y. Koike, E. Vatikiotis-Bateson, and M. Kawato, *Biological Cybernetics* **73**, 15–25 (1995).
14. J. K. Witt, and D. R. Proffitt, *Journal of Experimental Psychology: Human Perception and Performance* **34**, 1479–1492 (2008).