AGI Architecture Measures Human Parameters and Optimizes Human Performance

András Lőrincz and Dániel Takács

Eötvös Loránd University, Budapest H-1117, Hungary
andras.lorincz@elte.hu, dtakacs@gmail.com,
http://nipg.inf.elte.hu

Abstract. AGI could manifest itself in human-computer interactions. However, the computer should know what is on the mind of the user, since reinforcement learning, the main building block of AGI, is severely spoiled for partially observed states. Technological advances offer tools to uncover some of these hidden components of the '*state*'. Here, for the first time, we apply an AGI architecture for the optimization of human performance. In particular, we measure facial parameters and optimize users' writing speed working with head motion controlled writing tool. We elaborate on how to extend this optimization scheme to more complex scenarios.

Keywords: AGI architecture, computer-human interface, reinforcement learning.

1 Introduction

AGI developments face the problem of how to measure and compare achievements. The Turing test [5] and Turing games [1,10] could be good candidates. However, humans are 'equipped' with excellent evolution-tailored sensors to read the mind of the partner, develop models (make theory) about the others' mind. Such sensors and such theorization are also needed for AGI, especially if AGI aims human-computer interaction and collaboration, since reinforcement learning, the main building block of AGI, is severely troubled for partially observed states and so AGI is 'handicapped' without 'knowing' what is on the mind of the user. Here, for the first time, we use an AGI architecture for learning human parameters and for the optimization of human performance. AGI components comprise of (i) a sensory information processing system, (ii) a system that estimates the user's autoregressive exogeneous (ARX) process in the context of the actual goal, (iii) the inverted form of the ARX process for influencing the situation, as well as (iv) event learning [13] and an optimistic initial model [11] to optimize long-term human performance. In the experiments we measure parameters of the face in real time explore the relevant part of the parameter space and exploit that knowledge in head controlled writing using writing tool Dasher [14].

The paper is built as follows. We review our architecture in Section 2. Then we detail the experimental setup (Section 3). Results are presented in Section 4.

J. Schmidhuber, K.R. Thórisson, and M. Looks (Eds.): AGI 2011, LNAI 6830, pp. 321–326, 2011.

2 Architecture

At a very high level, the architecture is made of the following components: sensory processing unit, control unit, inverse dynamics unit, and decision making unit. Although it looks simple, one has to worry about a number of things, such as the continuity of space and time, the curse of dimensionality, if and how space and time should be discretized, and planning in case of uncertainties, e.g., in partially observed situations, including information about purposes, cognitive, and emotional capabilities of the user. A detailed description of the proposed architecture has been provided in [6] and some architectural components are under development. A simplified version of the architecture has been used for illustration; the algorithm learned to optimize the motion of a pendulum from raw visual information of 40,000 dimensions [7]. Below, we review the components of the architecture together with the present state of our developments:

Sample selection: This stage selects different samples under random control.

Low-dimensional embedding: In the illustrations, the dimensions of the low dimensional manifolds are known. Selected samples were embedded into the low dimensional space and were used for interpolation.

Identification of the ARX process: Out-of-sample estimations can be used for the identification of the ARX process [7] and Bayesian interrogation can speed-up the learning process [8].

Control: The inverted ARX process can be used for control both in the under-controlled and over-controlled cases.

LQR solution to the inverse dynamics: A demonstration was designed for the under-controlled situation. We used a linear-quadratic regulator (LQR) [7].

Event learning: Reinforcement learning (RL) has been rewritten into a novel event learning formalism [13] in order to connect RL and continuous control.

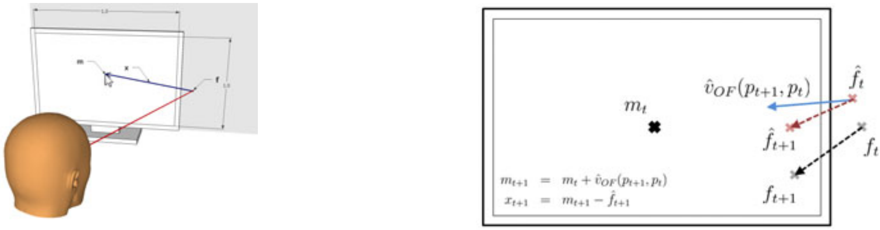
Exploring space and learning states: Optimistic initial model (OIM) was used for exploring the space and for learning the values of *events*. OIM suits large RL problems since it extends to the factored case [12].

Here, we use sophisticated preprocessing of sensory information (the pose of the head) before invoking components of the AGI architecture. At this stage of development we included domain knowledge. Below, we detail the architectural components used in the experiments.

Typical pose estimations use PCA methods for shape, texture, and details, see, e.g., [2,9] and references therein. We needed larger pose angle tolerance than offered by the presently available open source solutions and used a commercial software[1]. We used our Viola-Jones face detector and flow field estimation for detecting the face and relative changes of the head pose, respectively[2]. Input to the learning algorithm was hand made: Let us denote the screen size normalized position of the cursor by $(m_x, m_y) \in [0,1]^2$ and the head pose by $(f_x, f_y) \in \mathbb{R}^2$. The two-dimensional vector $x = [m_x - f_x, m_y - f_y]^T \in \mathbb{R}^2$ corresponds to the state (Fig. 1).

[1] FaceAPI `http://www.seeingmachines.com/`

[2] `http://chacal.web.elte.hu/.MouSense/MouSenseSetup-1.1.exe`



(a) (b)

Fig. 1. Illustration and parameters of the experiments. (a): Experimental arrangement. m: cursor position, f: position where the roll axis of head pose crosses the screen. x: 'observation'. (b): True and estimated quantities at time t. $\hat{v}_{OF}(p_{t+1}, p_t)$: optic flow based estimation of the motion vector f. p_t: positions of feature points on the 2D projected face at time t. For the definition of feature points see Fig. 2(a).

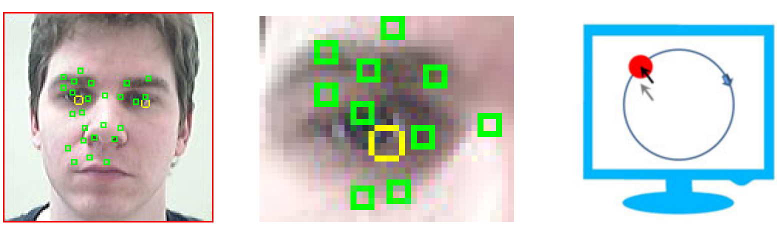
ARX estimation and inverse dynamics. The AR model assumes the following form

$$x_{t+1} = m_{t+1} - f_{t+1} \tag{1}$$

$$m_{t+1} = m_t + v_t \tag{2}$$

where $m_t \in \mathbb{R}^2$ is the position of the cursor at time t, $f_t \in \mathbb{R}^2$ is the point where the roll axis of the pose hits the screen as shown in Fig. 1(b), $v_t \in \mathbb{R}^2$ is the speed vector of the projected f_t on the screen over unit time and no additional noise was explicitly assumed. We have direct access to the cursor position and need to estimate the other parameters. Since $v_t = f_{t+1} - f_t$ it follows that $x_{t+1} = x_t$ in the absence of estimation errors and control. The goal is to control and optimize x_t for writing speed.

We do not have direct access to f_t or x_t, but use their estimations $\hat{f}_t \in \mathbb{R}^2$ and $\hat{x}_t$ through the measurement of the optic flow (Fig. 1(b)) of the face on subsequent image patches $(\hat{v}_{OF}(p_{t+1}, p_t))$, $p_t \in \mathbb{R}^{2k}$ denotes the 2D coordinates of k characteristic points (Fig. 2(a)) within the facial region of the image



(a) (b) (c)

Fig. 2. Tools for human-computer interaction: (a)-(b): features for optic flow estimation (green markers), eye tracker and head pose estimation (yellow marker), (c): computer 'game' designed for measuring the ARX parameters of head motion

Collecting a number of data $\hat{x}_1, \ldots, \hat{x}_T$, we estimated the unknown parameters of matrix B by direct control, using distances on the screen as $\hat{x}_{t+1} = \hat{x}_t + Bu_t + n_t$ and then inverting it to yield desired state x_d: $u_t = \hat{B}^{-1}(\hat{x}_d - \hat{x}_t)$. Inserting the result back to the ARX estimation we get $\hat{x}_{t+1} \approx \hat{x}_d$. We note that the inverse dynamics described here can be extended to sophisticated non-linear 'plants' [4].

Event learning. We define the optimal control problem within the *event learning* framework that works with discrete states, provides the actual state and desired successor state to a backing controller and the controller tries to satisfy the 'desires' by means of the inverse dynamics. For a given experienced state i and its desired successor state i^+, where $i, i^+ = 1, 2, \ldots, N$ and N is the number of states, that is, for a desired event $e(i, i^+)$, the controller provides a control value or a control series. The estimated value E^{π}_{i,i^+} of event $e(i, i^+)$ in an MDP is its estimated long-term cumulated discounted reward under fixed policy $\pi = \pi(i, i^+)$, Then, event learning learns the limitations of the backing controller and optimizes the RL policy in the event space [13].

Optimistic Initial Model (OIM). OIM extends the optimistic initial value method (OIV) [3]; it resolves the exploration exploitation dilemma by boosting with the 'Garden of Eden' state *and* by building a model. OIM brings about the optimal policy [11].

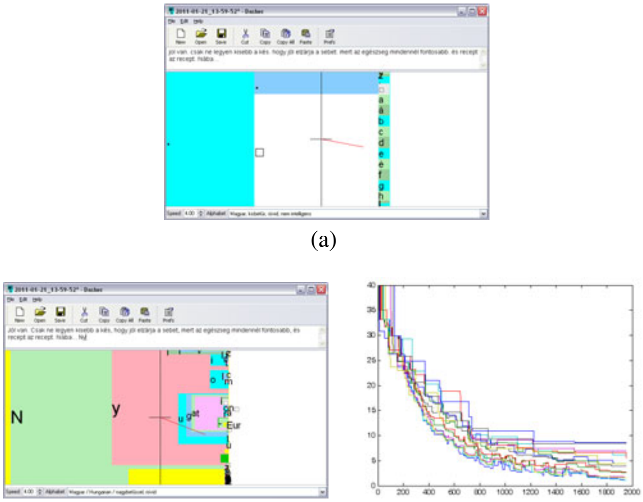
3 Experiments

Beyond the optic flow based head motion detector, we used the FaceAPI SDK for head pose estimation. A calibration procedure was applied at the beginning of the experiments. The principle is shown in Fig. 2(c): a red dot was moving on a circular path on the screen. The user could move the cursor by moving its head. The task was to keep the cursor within the red dot. If the cursor was within the dot then it speeded up, otherwise it slowed down. During the experiments, random control values were added to the motion in order to estimate matrix B. As expected, a close to diagonal matrix was learned with relatively small off-diagonal elements, being about one fifth of the diagonal values. Diagonal elements corresponded to the scaling (normalization) in the horizontal and vertical directions. This calibration was sufficient to learn the ARX process and to estimate the inverse dynamics.

The Optimistic Initial Model was configured as follows:

State space: Discretized differences between 2d cursor position and 2d crossing point of roll axis of head pose on the screen in pixels of a 1280×960 pixel screen
- horizontal direction: five regions with centers at (-512, -256, 0, 256, 512)
- vertical direction: three regions with centers at (-320, 0, 320)

Duration of time steps: 5 s.
Reward: Number of typed letters minus number of deleted letters during time steps.
Actions: Either the actual state itself, or one of the neighbor states (maximum number is 4) was chosen as the desired state. Inverse ARX was applied to move to the desired state (i.e., to modify the angle between the direction of the cursor from the head and the direction of the roll axis of the head pose).



(a) (b) (c)

Fig. 3. Dasher without (a) and with (b) prediction by partial matching. (c): convergence during learning [14]

Performance optimization was conducted with the Dasher writing tool with head motion controlled cursor movements (Figs. 3(a), 3(b)). For details on Dasher, please, consult [14]. We used two versions: the version with uniform probabilities for each letter (Fig. 3(a)) and the 'intelligent' version that utilizes the method of prediction by partial matching and scales letter areas according to their estimated probabilities (Fig. 3(b)). Two subjects conducted the experiments. Experiments lasted for five sessions, each session had five writing periods and four breaks between, with and about 600 characters to be typed in each period. Periods took about 20 minutes.

4 Results and Conclusions

Convergence of optimization is shown in Fig. 3(c). Optimal policy did not place the cursor to the crossing point of the screen and the roll axis of the head for either subjects. Furthermore, we got different optimal policies for the two subjects. Intelligent Dasher gave rise to cca. a factor of 2 better performance for both subjects.

In summary, we have used an AGI architecture [7] for the optimization of human performance during head pose driven writing. We combined visual and textual information, we identified the dynamics of head motion, used the inverse dynamics to control the direction of head and optimized performance by means of reinforcement learning methods like event learning [13] and the optimistic initial model [11]. The architecture is scalable in most aspects, including sensory information processing, robust control, and reinforcement learning [7]. The relevance of the present study is that we used tools to measure human parameters, coupled those to an AGI architecture and used the architecture to improve human performance.

Acknowledgments. Research was supported by the European Union and co-financed by the European Social Fund (grant no. TÁMOP 4.2.1./B-09/1/KMR-2010-0003).

References

1. Berman, J., Bruckman, A.: The Turing game: Exploring identity in an online environment. Convergence 7, 83–102 (2001)
2. Cristinacce, D., Cootes, T.: Automatic feature localisation with constrained local model. Pattern Rec. 41, 3054–3067 (2008)
3. Even-Dar, E., Mansour, Y.: Convergence of optimistic and incremental Q-learning. In: Advances in Neural Information Processing Systems, vol. 14, pp. 1499–1506 (2001)
4. Khalil, H.K.: Nonlinear Systems. Prentice Hall, NJ (2002)
5. Loebner, H.: How to hold a Turing test contest. In: Parsing the Turing Test, pp. 173–179. Springer, Netherlands (2009)
6. Lőrincz, A.: Learning and representation: From compressive sampling to the 'symbol learning problem'. In: Handbook of Large-Scale Random Networks, pp. 445–488. Springer, Heidelberg (2009)
7. Lőrincz, A., Bárdosi, Z., Takács, D.: Sketch of an AGI architecture with illustration. In: 3rd Conf. on Artif. Gen. Intel. (2010), `http://dx.doi.org/10.2991/agi.2010.40`
8. Póczos, B., Lőrincz, A.: Identification of recurrent neural networks by Bayesian interrogation techniques. J. of Mach. Learn. Res. 10, 515–554 (2009)
9. Saragih, J., Lucey, S., Cohn, J.: Deformable model fitting by regularized landmark mean-shifts. Int. J. Comp. Vision (in press)
10. Schaul, T., Togelius, J., Schmidhuber, J.: Measuring intelligence through game. J. Artif. Gen. Intell. 2, 1–19 (2010)
11. Szita, I., Lőrincz, A.: The many faces of optimism: A unifying approach. In: 25th Int. Conf. on Mach. Learn., pp. 1048–1055. Omnipress (2008)
12. Szita, I., Lőrincz, A.: Optimistic initialization and greediness lead to polynomial time learning in factored MDPs. In: 26th Int. Conf. on Mach. Learn., pp. 126–133. Omnipress (2009)
13. Szita, I., Takács, B., Lőrincz, A.: Epsilon-MDPs: Learning in varying environments. J. Mach. Learn. Res. 3, 145–174 (2003)
14. Ward, D.J., MacKay, D.J.C.: Fast hands-free writing by gaze direction. Nature 418, 838 (2002)