A Design for Living Technology: Experiments with the Mind Time Machine

Takashi Ikegami*
University of Tokyo

Keywords
Mind time machine, default mode network, autonomy, chaos, video feedback, neural network

A version of this paper with color figures is available online at http://dx.doi.org/10.1162/artl_a_00113. Subscription required.

Abstract Living technology aims to help people expand their experiences in everyday life. The environment offers people ways to interact with it, which we call *affordances*. Living technology is a design for new affordances. When we experience something new, we remember it by the way we perceive and interact with it. Recent studies in neuroscience have led to the idea of a *default mode network*, which is a baseline activity of a brain system. The autonomy of artificial life must be understood as a sort of default mode that self-organizes its baseline activity, preparing for its external inputs and its interaction with humans. I thus propose a method for creating a suitable default mode as a design principle for living technology. I built a machine called the *mind time machine* (MTM), which runs continuously for 10 h per day and receives visual data from its environment using 15 video cameras. The MTM receives and edits the video inputs while it self-organizes the momentary *now*. Its base program is a neural network that includes chaotic dynamics inside the system and a meta-network that consists of video feedback systems. Using this system as the hardware and a default mode network as a conceptual framework, I describe the system's autonomous behavior. Using the MTM as a testing ground, I propose a design principle for living technology.

1 Introduction

Any basic science can lead to innovative applications. Artificial life (ALife) studies are no exception. The purpose of living technology is to bring to fruition the concepts developed through the study of ALife, such as self-reproduction, autonomy, enaction, robustness, open-ended evolution, and evolvability, in a real-world context [15].

In this article, I will discuss a design principle of living technology through a real-world ALife system. In contrast to an artificially simulated environment, the real world presents many unexpected complex encounters, and living systems are essentially adaptive to these complexities. I designed an artificial system that can be a first test of artificial systems' ability to survive autonomously in an open environment. We require that any ALife should be autonomous in the first place and cope with various kinds of sensory flows while simultaneously maintaining its own identity and autonomy over a long period of time.

* Graduate School of Interdisciplinary Information Studies, the University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan. E-mail: ikeg@sacral.c.u-tokyo.ac.jp

In previous studies on ALife in the real world, I conducted novel biochemical and robotic experiments; these approaches are being used to develop an in-depth understanding of robustness [16, 18, 27]. Oil droplets demonstrate a simple chemical phenomenon in response to high pH: Water reacting with oleic anhydride generates self-moving droplets that maintain the reaction on its surface, sustaining self-mobility [4, 5]. Here, the environmental conditions, pH and oleate concentration, are controlled by the droplet motion. I define the robustness of the droplets in terms of their ability to sustain self-moving behavior. In contrast, if we pick an example from the game of *Life*, we observe that gliders, the simplest moving pattern in the game, appear to display self-moving behavior, but do not actually function in that way. This evolution of self-movement, autonomy, and individuality appears to be a key prerequisite for developing robust behaviors [15].

Using a robotic platform, we used pure Hebbian learning dynamics [23] to show how auditory and visual modules work together to self-organize robust goal-oriented behavior [9]. When one gives a robot the capacity of self-organization, however, the robot cannot sustain its autonomous movement quite so easily. Developing robustness in this case appears to depend on the development of an appropriate use of time scales for its behavior; in particular, parameter settings for the robot's learning and forgetting during the process of Hebbian learning can affect the time scales of the robot's behaviors. Finding a range of these parameters that allows proper functioning by utilizing the background noise of the environment will allow for the development of more robust behaviors.

By increasing our understanding of how we can connect artificial systems with natural environments, we can further our development of a theoretical framework that situates ALife in an open environment. Putting aside self-movement, in this article I discuss a principle for designing living technology in order to implement autonomy in technology. Obtaining robust autonomous behavior is a prerequisite for ALife in the real world, but that doesn't mean that ALife ignores the environmental context, which is a mere independent behavior. However, ALife must become sensitive to the environment, including human beings in the environment. Hence, the definition of autonomy here is the self-determining behavior of ALife that utilizes past experiences and future predictions.

A mechanism of autonomous behavior may be attributed to a mode (or a baseline activity) of a system while it receives no input from the outside. In human brain systems, a default mode network is proposed as a special neural activity that is responsible for the baseline activity of a brain without any specific task [2, 24, 25]. I generalize the notion of the default mode to any autonomous system. Living technology aims to help people to expand their experiences in everyday life; that is, it is a design for new affordances. My hypothesis is that preparing a default mode is necessary for both perceiving and creating new affordances, even with ALife.

The example of a living technology for this article is the *mind time machine* (MTM) [13]. I built an MTM that runs continuously for 10 h per day and receives visual data from its environment using 15 video cameras. The MTM receives and edits the video inputs while it self-organizes the momentary *now.* Its base program is a neural network that includes chaos dynamics inside the system, and a meta-network that consists of video feedback systems. Using this system as the hardware and the default mode network as a conceptual framework, I describe the system's autonomous behavior in the above sense (cf. the different arguments concerning the same MTM in [13]). Then, I argue that the MTM can provide a testing ground for developing living technology.

In order to avoid confusing living technology with the classical AI approach (the MTM does not have actuators), I compare the essentials of living technology and AI here. In general, an AI is a symbol-based, fully determined program that has an explicit goal to achieve. For example, general machine-learning techniques, including neural networks and evolutionary computation, are typical examples of the current AI approach to producing intelligent systems. On the other hand, living technology is an application of ALife, that is, of autonomous embodied self-(re)producing systems. One key difference from the classical AI concept is that in ALife studies intelligence is only taken to be a side effect of life systems. The primary purpose of living technology is not to optimize things but to sustain itself. The MTM uses neural networks and no explicit actuation. Its objective is to survive in an open environment, without losing sensitivity to the environment. In this sense, the MTM is a prototypical example of living technology. It also provides an example of a *metadynamical system* in that its parameters and time step are not predefined.

I named MTM after Libet's book *Mind Time* [19], since MTM self-organizes its own state update timing through one of its variables. This in turn is affected by its memory and input coming from the outside environment. Libet argued that our subjective temporal order does not have to coincide with the actual temporal order of events in physical time. Our brain system "corrects" the temporal order by referring backward in time in order to give priority to the primary evoked response of the sensory cortex to the stimulus. We will come back to this point in Section 4.4, where the system's subjective time is discussed.

In Section 2, I present the background that is relevant to understanding the MTM. The notion of a default mode is introduced. Then, in Section 3, I introduce the architecture of the MTM. In Section 4, I report how the MTM behaved over 2.5 months and characterize the behaviors, such as its temporal complexity and the dynamics of its internal states. In Section 5, I discuss how the MTM's autonomous behavior can be interpreted in terms of a default mode network. We then return to the notion of living technology at the end of the article.

2 Related Background

Bringing ALife into the real world means letting humans interact with ALife, not only visually, but also physically. A good example of this was Sony's puppy robot AIBO, which people could interact with as if it were a real dog (http://www.sony.jp/products/Consumer/aibo/index.html). There was a nice story about a girl who came to see AIBO and was pleased that AIBO always chased her, but what actually happened was that AIBO was designed to chase the color red.

Though AIBO was a large commercial success, its production was ended in 2003. It attracted people because it could learn and adapt to its owner. In order to attract people, it is necessary to have unexpected behavior. Oudeye and Kaplan [23] implemented a meta-learning circuit algorithm, in which they tried not only to reduce the prediction error, but also to increase the temporal changes of the error rate—that is, to provide for getting bored with the fixed pattern. This might create a new type of affordance for AIBO. It is shown by Scheutz [26] that a small amount of selfish agency—for example, a robot that monitors its own energy consumption and puts a priority on this—will be good for human-agent interaction. The only problem with those robot examples is that their appearances are already very much like those of existing animals, such as dogs or cats. In such a case, we can make, at most, animals similar to dogs or cats.

Creating a living technology means essentially providing a new affordance (a pattern in the environment that affords people the ability to interact with it), which we have not yet experienced. When the Internet appeared, people did not know how to use it. Many people do not know how to interact with contemporary art, but some of the arts have established better relationships between people. A key to having a better relationship can be found in the quality of affordance. One purpose of making living technology that utilizes ALife concepts should be creating new types of affordances.

The perception of an affordance requires us to prepare for sensory-motor couplings. Namely, the perception of an affordance automatically and unconsciously leads to the production of motor outputs; for example, a visual appearance of a glass shape induces a hand motion for grasping the glass. In order to accomplish such automatic organization of motor outputs, a brain system must prepare a *readiness state* in advance that will resonate with the affordance.

My hypothesis here is that a readiness state can be attributed to a default mode of a brain system [2, 24, 25]. A brain region that is responsible for a given task is identified as one that has higher activity than the baseline. A natural question comes from Buckner et al. [2]: What is the baseline neural activity itself, and how can we measure it? They have studied the baseline activity by studying the regions that become less active when a specific task is given. This successful approach uncovered some remarkable perspectives on the default mode. Among them, according to the reviews by Buckner et al. [2] and also Raichle et al. [25], are:

(i) The area associated with the default mode is an integration of various subsystems in the brain; the medial prefrontal cortex and posterior cingulate cortex subsystems seem to play a central role.

(ii) The neural activity of those subsystems consists of noisy fMRI signals at a low frequency of about 0.1 Hz or less, showing global synchronization.

(iii) The default mode has to do with spontaneous cognition (e.g., daydreaming; internal thoughts, including planning for the future).

(iv) The activity of the default mode is anticorrelated with that of other brain regions that are responsible for focusing attention on the external world.

(v) The brain region associated with the default mode overlaps those manipulating the episodic memory.

A brain switches temporally between the default mode, which is responsible for spontaneous cognition, and the reactive mode, which is responsible for reacting to external stimuli. This notion of the default mode and thus the switching between the default and reactive modes can be generalized to any living systems with or without brains.

I think that synthesis and the perception of affordances have a common root. We need a default mode to perceive affordances, but we also need it to create a human-agent interaction. Therefore, the autonomy of ALife must be understood as a sort of default mode that self-organizes its baseline activity, preparing for the external inputs and interacting with humans. I thus think any kind of living technology needs its own default mode. The way to make a suitable default mode is a design principle for living technology. In the following section, I summarize the work of the MTM as an example of living technology. Based on what the MTM does, I will discuss its default mode.

3 Mind Time Machine

3.1 Perception of Visual Images

I presented the MTM as an art project in the indoor-outdoor space at the Yamaguchi Center for Arts and Media (YCAM) in March of 2010. While demonstrating this MTM at the museum, I transferred data for analysis from the machine every day. The system's initial states were reset every morning, but its long-term memory accumulated over 67 days.

The MTM consists of three module subsystems, and each has a projection screen. Those screens are set to its right and left and above it, sharing the corner of a cubic skeleton with a 5,400-mm width on each side (see Figure 1). Fifteen video cameras (one of which was used for detecting whether anyone was in the venue) were attached to each pole of the skeleton, in order to view things happening in the venue. The movie images are decomposed into frames; I designed four distinct ways to recombine the frames into a new frame, as described below.

This visual scheme is motivated by the fact that visual information propagates from the retina to the primary visual cortex, which already receives many projections from the cortical regions (e.g., [3, 28]). Accordingly, the simultaneity of the visual bits will be violated as they propagate further and further into the higher-order cortical regions. I assumed that the visual inputs for MTM should be twisted and edited in a similar manner, reflecting the real visual pathways.

Here are the kinds of visual effects I used:

(1) slit-scanning movies (picking up a vertical pixel line from different time frames to generate one image sequentially);

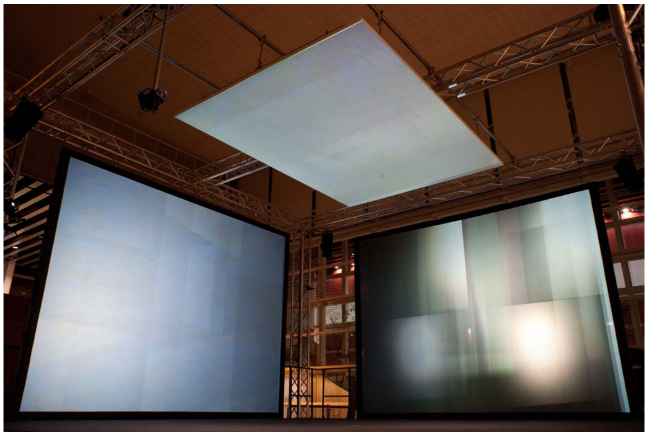

Figure 1. View of the MTM displayed at the Yamaguchi Center for Arts and Media, 2010 (photo taken by Kenshu Shintsubo).

(2) superposing images from different time frames (I used the simplest α-blending method to do this);

(3) reversing the time order of frames to make a reverse-time movie;

(4) video feedback (dividing an input into 2^k regions—the number of divisions changed with time—each of which contained the same image). The video then takes the images of that divided screen, so that it inevitably generates self-similar images, except when there is no division.

For each time step t, a visual image from a camera is superimposed on those from another one or two cameras to generate a new mixed image frame. I made m different combinations of images $I_k(t)$ ($k = 1, 2, \ldots, m$), and those mixed images were α-*blended* (superimposed with the weights α_k) to generate the combined image $I^*(t) = \sum \alpha_k I_k(t)$. This $I^*(t)$ will be stored in a subsystem as a memory of the environment at that moment t. As a result, the MTM "experiences" the environment as $I_w = \{\ldots, I^*(t-2), I^*(t-1), I^*(t), I^*(t+1), \ldots\}$.

We assume the time series I_w is an internal image of the environment stored by the MTM (stored by the neural weight, which will be described in the next section), and we have three such different images, I_{w0}, I_{w1}, and I_{w2}, corresponding to three projection screens (see the sketch in Figure 2). What will be projected on each screen is, however, not the I_{w0}, I_{w1}, or I_{w2} itself. We let a subsystem select one image, $I_k(t)$, out of the k input patterns that have the largest α_k value (α-blending coefficient). Therefore, what one can see at time t on the three screens are the images that have the largest α for each of the three corresponding displays, that is, the three subsystems.

The three subsystems are updated asynchronously and are coupled only optically, in the sense that cameras for making visual images on screen 1 will shoot screen 2, and so on. Subsystem 0 has image inputs from passive cameras that shoot screens 1 and 2. Subsystem 1 has image inputs from the active cameras, which shoot its own screen and screen 2. Subsystem 2 has no direct inputs from cameras, but does have inputs from the input directories from subsystems 0 and 1, namely, their projection images on the screens, and also old memory images that subsystem 2 has displayed before. Therefore, subsystem 2 does the α-blending of only three input images.

The next stage is updating the α-values, for which I used a double-layered neural network.

3.2 Plastic Neural Networks

I used a fully connected neural network for each of the three subsystems to process the images in the previous section. The network consists of two layers, and the dynamics obey the usual integrated and firing dynamics and the modified Hebbian dynamics [6]. The purpose of using the network is to

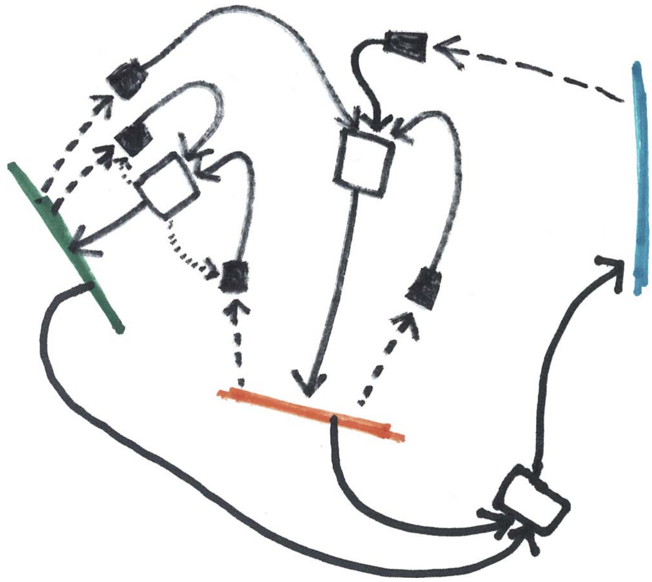

Figure 2. A sketch of three screens and the video cameras. Dashed lines indicate that a camera captures the corresponding screen image, and solid lines show the transportation of the images from each camera to computers and their projection from the computer onto a screen.

memorize the input images, determine the α weight of each image, control the camera movements, and update the screen images.

Images received from the video cameras are progressively embedded into the neural connections of the first layer via the Hopfield method [7, 8]. Connections from the first layer to the second layer are progressively self-modified according to modified Hebbian dynamics, that is, the connection strength is modified in proportion to the correlation between the presynaptic neuronal state and the postsynaptic neural state change. The connections between neurons in the second layer are also modified by the same dynamics.

Each neuron in the second layer is associated with a set of visual inputs, and the set associated with the most activated neural state will be selected for projecting onto the output screen. This is a winner-take-all rule. However, non-selected sets of images are also superimposed on the neural weight. This was intended to reflect the idea that perception is essentially working in parallel. Non-selected modes should also influence the makeup of the system's memory pattern.

By fixing the memory-embedded connections, every neuronal state is updated by particular neural dynamics, which are known as chaotic neural dynamics [1, 20]: By changing the connection strengths, we see that a network demonstrates fixed-point, periodic, and chaotic behaviors. Each neuron in the second layer is associated with a set of visual images, and a simple winner-take-all rule is applied here. Namely, a combination of visual images corresponding to the largest neural state is selected to be projected onto the screen.

Visual images are acquired and replayed in a recursive way (Figure 3). The system itself is a completely deterministic one, which uses no random numbers, but it projects different images depending upon the inherent instabilities of the neural dynamics that reflect environmental light conditions, the movement of people coming into the venue, and the system's stored memory. Each neural state, as well as synaptic strength, is updated Q times (we took $Q = 10$ in this experiment) for each real-time duration T. Here T ranges from 0.1 s to 1 s and is determined by a designated neural state of the second layer.

The procedure is as follows: (1) Cameras take the visual images, and (2) those are accumulated in the weight. (3) The weight value determines the bifurcation of dynamics, and (4) the associated neural states determine which image should be projected onto a screen. These steps are repeated all day (Figure 4).

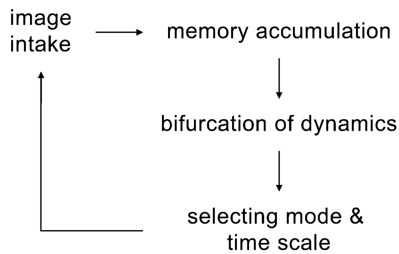

Figure 3. A simple block diagram for how the system is processed iteratively.

3.3 Some Other Remarks on the MTM Setup

The system consists of three subnetworks (net0, net1, net2).

(1) Subsystem net0 is tentatively called the *unconscious* module because its inputs come from 10 passive video cameras. The camera positions are fixed in space.

(2) Subsystem net1 is called the *conscious* module because its inputs are from two active and two passive video cameras. The parameters for the movement of the active cameras are also determined by the neural states of the second layer.

(3) Subsystem net2 has no camera inputs. Instead, its inputs come from screen 0 or 1 or from the buffer, which accumulates long-term images of its own. Images from subsystem net0 are sent to screen 1 (visualization of unconscious states: passive images), and those from subsystem net1 go to screen 2 (visualization of conscious states: active images). Images from subsystem net2 are sent to screen 3, which we call episodic memory: bundles of previously produced images from either the conscious or the unconscious screen.

Most cameras are shooting screen images, so video feedback will readily occur. Most cameras are zooming the screen images, so they sometimes induce oscillation excitation. It also happens that the same image pattern is circulated for a while among the three screens due to a mutual feedback loop, that is, one camera shoots the screen from the other camera, its own screen is shot by the other camera, and so on.

The network is fully plastic and changes its structure over time. This network partially "remembers" the previously received inputs, but this does not mean that the subsystem can stably retrieve the memory. In fact, we know that neural dynamics with a fixed connection strength also produce complex and temporally unstable behaviors, such as chaotic itinerancy [10, 11, 14, 17, 20]. This network copies the spatial-temporal correlation that exists in its environment into the correlation of synaptic weights within the network. One assumption at work here is that perceiving something requires

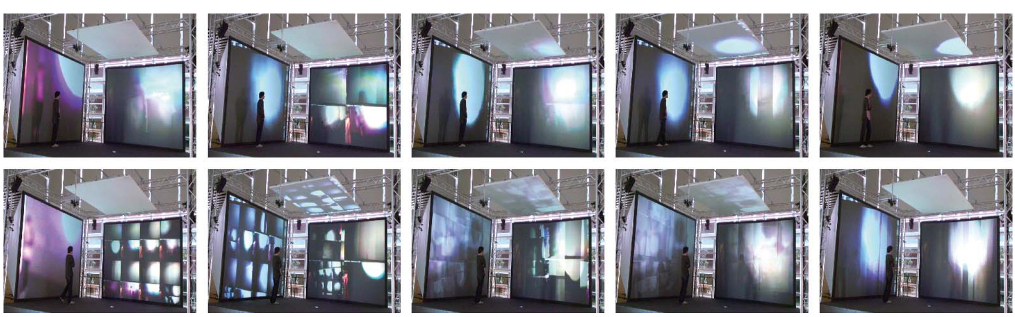

Figure 4. An example of how screen images change over about 10 s.

changes in the functional state of the underlying neural dynamics, namely that the network parameters are changed by the images they receive, so the perception of images often goes through dynamic bifurcations. In particular, what a subsystem has perceived before determines what to perceive next. The above architecture reflects this hypothesis.

When people come into the venue, their images are captured by the video cameras, and are then recognized and processed by the neural dynamics. Intake images are explicitly affected by human movements, but also by the projection mode. The four projection modes detailed in the previous section have the following effects: Movements are strangely stretched or contracted in the images. For example, when a person tosses balls into the air, the number of balls increases. The person osberves being replicated in the video. The person can be seen moving backward, in reverse time. Bodily movement is embedded in the screen at different scales. Those effects certainly affect the MTM's behavior as people come to see and interact with it, yet quantitative analysis of these effects is a problem to be explored in the future.

In addition, I generated sound images by scanning each screen. I raster-scanned (swept the two-dimensional image horizontally step by step) the frames successively, translated them into sound amplitudes, and played the sound by filtering them through a prepared frequency spectrum. In this way, the dynamics of the visual images are translated into sound patterns, which also affect the human observers. The generated sounds thus covary as the visual images change with the programs that stretch, fold, and superpose.

4 Observation and Analysis

Rather than a mere chaotic system, I think that the MTM is an artificial "living" system, because the observable behavior of the MTM is due to its ever-changing, mostly transient processes, rather than its attractor organization. The MTM can only set its parameter set once it is put in the environment. In this sense, the MTM becomes a metadynamical system that creates a dynamical system with environmental patterns as its input. However, my main concern is that we examine the MTM as a real-life pet. Specifically, we should not analyze its internal mechanism, but first *observe* the MTM's daily behavior by creating a diary.

In sum, three important remarks will be made here (see also Figure 3):

(1) The MTM will persistently tune the parameters of its dynamics by using the images that it captures, and accordingly, the associated attractors will change.

(2) The MTM changes the values of network connections to accumulate visual images sequentially, thus constituting an *episodic memory.*

(3) The MTM regulates and maintains the intrinsic dynamics even when nothing special happens in the environment; we call this the *default mode* of the MTM. When nobody enters the space, the MTM repeatedly projects the empty space through the screens, and as a result, the same place is engraved on the weights of the network. Since the MTM was installed in a half-open space, the sunlight changes from morning to evening, so the weights can be modified by the light conditions. Also, the third screen sometimes changes by recalling long-past images from the web, which may perturb the weights. Therefore, the default mode is not characterized by a single attracting state of fixed parameters, but by a set of dynamical states. Figure 4 depicts an example of how the MTM images change over time.

In order to track the behavior of the MTM, I recorded neural weights, states, and screen images, with which we can plot: (1) a return map of the successive neural states for all day long, (2) the dynamics of the weight strengths, their average values, and their standard deviations, (3) temporal changes of the video images, and (4) the internal memory updating, and so on.

4.1 Return Map Analysis

I define a *return map* of the neural states as the superposition of the two successive neural states on one figure (Figure 5). The importance of a return map is that the overall dynamics are projected onto a two-dimensional plane so that it generally becomes a blur, but it sometimes falls into a few points, implying that the dynamics are in a periodic state.

By producing return maps every 2 h, I observed how the return map varies over time and that the pattern sometimes shrank into a few points, completely disappeared, or blew up. As for the general tendency, subsystems net0 (with fixed cameras) and net2 (memory images) easily lose their complexity in return maps, but subsystem net1, with mobile cameras, can sustain complex dynamics. Still, subsystems net0 and net2 also recover rich patterns after a few hours. Once the three of them lose their complex pattern, they all become silent, which occasionally happened. A day-long homeostasis is often brought on by this net1 dynamics.

4.2 Taking a Daily Diary

I made a daily diary of return maps (Figure 6) to see how they change over time. The maps look similar, but not exactly the same. Some return maps were more complex than others, and some days they became totally silent.

In order to discover what controls the behavior of the MTM, I examined the weather conditions during the time the MTM was displayed. I then found that the return maps always became quiet when it rained all day. On the other hand, there were quiet return maps even when the weather was fine.

Since the MTM receives its visual inputs from the venue, this changes the dynamics of the neural network. Viewings of the same pattern converge into the same dynamics. In some cases, when a certain attractor is hit by the system, it will not recover the original state, so it eventually becomes quiet (sparse patterns in Figure 6). Whether it will recover by producing some effective visual inputs is not fully understood. In addition to the effect of weather conditions, the neural states also became quiet after the sun set. Low light intensity is the main factor that suppresses the neural state. This rather simple light condition also limits the MTM dynamics, but whether the venue is busy or not should be a potential factor. On Sundays, more children come to the venue, which may activate the MTM's dynamics.

4.3 Weight Change

The weight strength of the neural network plays the role of a parameter that controls the dynamic neural state. Too big or too small a weight strength means that the network has a stable fixed point, that is, it shows a temporally constant state. In Figure 7, the mean (red line) and the standard deviation

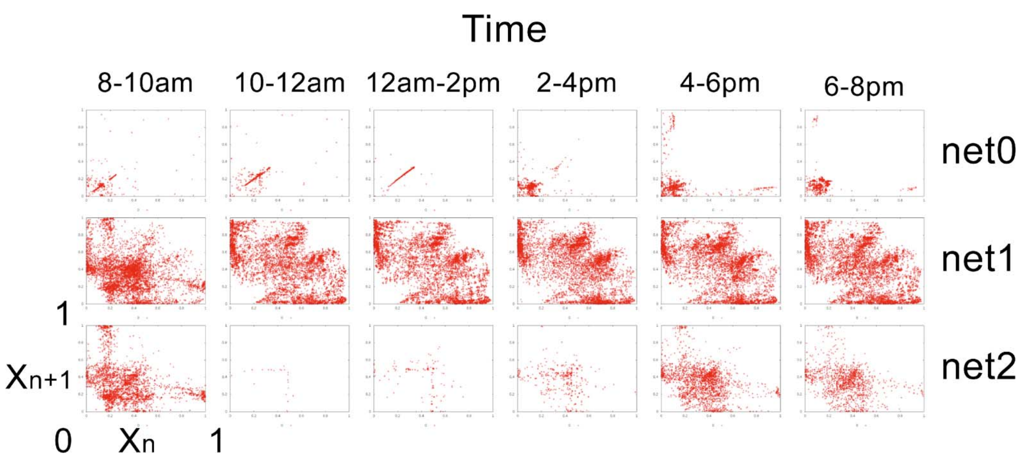

Figure 5. Temporal changes of return maps, plotting a neural state of successive time steps (X_n, X_{n+1}), by randomly picking up a neuron from each of the three subnetworks (data taken from April 7, 2010).

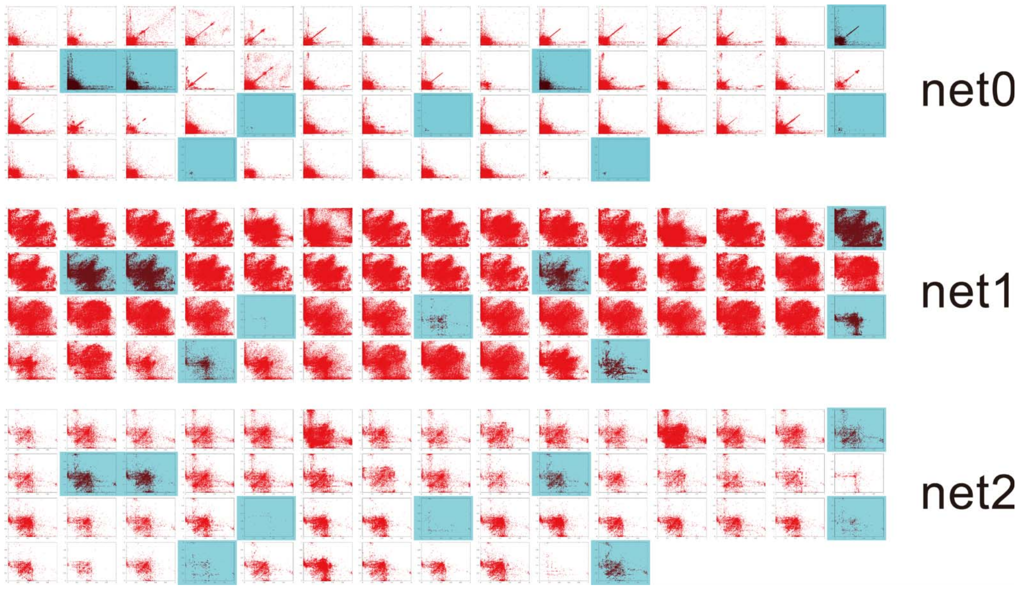

Figure 6. Daily superimposed return maps from each subnetwork net0, net1, and net2 for 56 days during the experiment. Each return map corresponds to a single day. In the online version, blue backgrounds indicate fully rainy days (no sunlight). As in Figure 5, each panel displays a return map of x_n versus x_{n+1}.

(green line) of each subnetwork are depicted. Only at intermediate values (around 0.25 and 0.75, due to the symmetric structure) does the network show complex dynamic behavior, such as long-term periodicity and chaos. I computed the weighted average and the standard deviation, tracking them from 8 A.M. to 8 P.M. every day to see what was responsible for complex behavior. I found that an increase in the variations of weight strengths in a network gave stable dynamics, but as in the lowest column of Figure 7, sudden collapses of the variation led to chaotic behavior.

4.4 The System's Subjective Time

The MTM organized its own time scales, which are driven by the memory structure. That is, the neural states were updated repeatedly (i.e., visual inputs memory organization → dynamics change → determining what to project); see Figure 3.

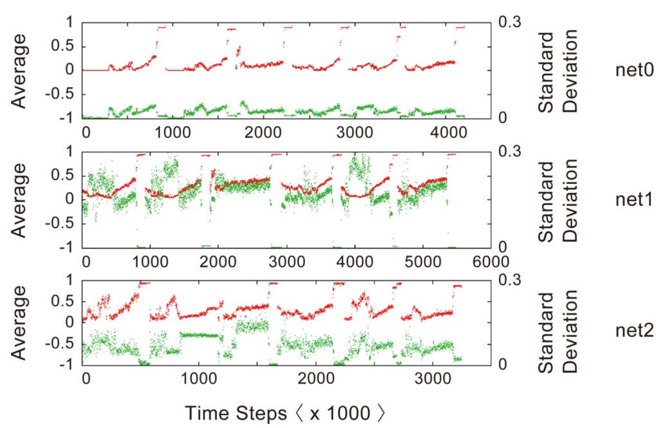

Figure 7. In the online version, temporal changes of average value (in red) and deviations (in green) of connection strengths for seven days for each of the three subnetworks. The horizontal axis is updating time steps. When the average strength reaches the maximum value (1.0), that marks the end of each day when the museum closes. How many times the states of the MTM update varies every day (see Figure 8).

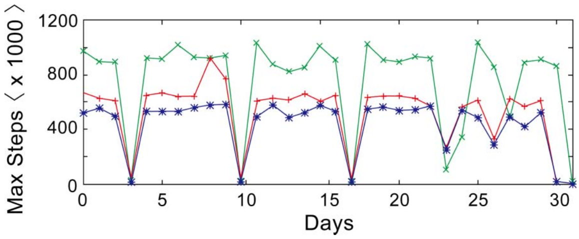

Figure 8. The total number of updates per day over the last four weeks. Depending on the daily conditions, the number changes significantly. It goes to zero when the museum is closed every Monday. Actual operation duration per day is about 12 h. Different colors correspond to the different subnetworks.

In this case, when to update the memory is determined simply by a neural state multiplied by the parameter TIME, which is assigned the value 1 s. If that neural state is close to zero, memory is frequently updated. On the other hand, if its value is large, memory is rewritten every 1 s at most. I take this memory-updating time scale to be a candidate for the system's subjective time scale, which is a direct outcome of neural dynamics, but indirectly a function of memory organization.

The neural state update time (subjective time), as a function of Newtonian time, sometimes goes slower than the physical time scale and sometimes faster. The 67-day experiment revealed that the total number of memory updates depended on the day (Figure 8). In particular, each screen had a different total number of updates, which also changed every day. I speculate that the daylight conditions also mattered in this accelerating pattern.

Libet [19] has argued in his *time-on* theory that sensory inputs must last at least 400 ms in order to convert unconscious correct temporal order to consciously experienced temporal order. The MTM's updating time is also primarily governed by the image that is most frequently projected onto a screen. In order to generalize the idea of a time-on theory to the MTM and general systems, I plan on investigating a more explicitly managed algorithm for backward referral in time.

5 Discussion

Why do we need autonomy in technology? This is the question we must now ask. What is unique about the MTM is that it can reconfigure its memory structure by using autonomous dynamics, that is, the video feedback and the neural dynamics, whose parameters are varied by storing the images. When this autonomous dynamic "dies," it means that the dynamic becomes a fixed point and never changes on further accumulating images. As we have seen in the previous sections, the MTM becomes quiet on rainy days or late in the evening.

I view this sustainable behavior with respect to falling down into the dead state as a *default mode* of the MTM. Like the default mode in a brain system, the default mode in the MTM is the baseline activity of the system. It self-sustains complex dynamics for maintaining, memorizing, retrieving, and reacting to environmental changes much as we expect the default mode of the brain system does. We expect that the default mode in the MTM makes it possible to predict future environmental changes. The existence of this default mode might make it easier to let the MTM self-organize, rather than requiring us to control it. That is why we need autonomy, even in artificial systems.

Let us compare the behavior of the MTM with that of the previous autonomous systems in ALife. We are concerned with what would be a relevant dynamic that introduces self-sustainability [15, 27]. Among our previous studies of ALife, self-moving oil droplets [4, 5] and Hebbian learning robots [9] provide examples of self-sustainability mechanisms, and the MTM adds another possibility. Do we see any common properties of these examples that could help us define a default mode in general?

In the case of the oil droplets, the droplets self-control their navigation behavior (when and where to move). As for external perturbation, if we insert a high-pH region, the droplet reacts to the pH and changes its behavior. We observe that a droplet is sometimes attracted to the high-pH region, but sometimes it avoids such a region. In the former cases, the behavior is not a function of the default mode, but indicates simple reactive action. In the latter ones it is more like a default intrinsic behavior that neglects the external pH. These two totally opposite behaviors may be related to the two activities of the default mode in a brain system.

It should be noted that the default mode is different from a mere deterministic mode, in which a system determines its behavior without having any external inputs. A deterministic autonomous mode is a product of a closed system, but the default mode is designed to organize autonomous dynamics by spontaneously ignoring some external inputs. Thus, the default mode can be said to characterize a self-organized dynamical system.

In the case of the Hebbian robot, we say that it is self-sustainable when a robot can self-navigate toward the goal point. When there exist only a few gradients of sounds or lights in an environment, a robot may change its navigation pattern randomly. This corresponds to the default mode of the robot. Once there is a sufficient gradient in the environment, the robot reacts to and follows the gradient. However, comparing it with the oil droplet experiment, we note that the response behavior is not a function of the default mode. It is more or less merely a reactive action.

Compared with those cases, the MTM has a more powerful default mode. The MTM sustains its baseline activity when nobody is interacting with it. When people come to interact with it, the MTM sometimes reacts strongly to them (so that the people see the effects of their motion patterns), but sometimes the reaction is diminishing. The response behavior depends on the dynamical state and the weight organization of the neural dynamics.

We may also find it interesting that the MTM tends to self-organize at intermediate time scales. There exist four fundamental time scales in this system. One is the neural state updating, which is the fastest. The second is that of memory accumulation in the first layer. The third is the Hebbian time scale, and the fourth is that of memory retrieval from the Web. What we mean by the production of an intermediate time scale is the spontaneous collapse of its weight distribution. Likewise, in the circadian rhythm, the MTM tends to have an own periodicity, namely, collapses in its weight value fluctuation occur two to three times a day (see Figure 7). However, this collapse is not observable every day. It may be another sign of the default mode of the MTM.

In addition to the MTM, another example of a default mode in an artificial system is the Web, one of the most developed artificial systems of our era. The Web responds to real-world events, which are measured by the numbers of queries and tweets, but even without any salient real-world events, the Web can be self-activated. We can consider the internal bursts, which are not driven by real-world events, but by the internal dynamics of the Web, as a default mode of the Web system [22]. By using the transfer entropy, I measured the mutual effect of tweet postings and Google queries. I analyzed the default mode by using the sink state, where information flow enters, but does not come out, and the source state, where information flow comes out, but does not enter the state. As in the discussion of real brain systems, the hypothesis is that the default mode can be an important Web mode, not only for supporting baseline activity but also for reducing uncertainty in information circulation on the Web, thereby regulating the consistency of information on the Web and in the actual world.

Concerning the above examples of ALife systems, I think that living technology should include a default mode. The default mode of the MTM is characterized by self-organization of the time scale and by self-sustained rich chaotic dynamic patterns. Not only the MTM, but also oil droplets and other ALife systems, possess primitive forms of the default mode. We expect these characteristics in all potential living technologies. Specifically, any living technology should have its own default mode (autonomy) in order to respond to the environment, and the default mode itself should be self-sustaining.

Living technology must be a new kind of technology that we will not initially understand how to use. Interacting with a system automatically organizes a purpose for the system. In case of the MTM and its sequels, living technology is a means to provide a novel form of human-agent interaction: not a face-to-face, but a spatially and temporally distributed one. For example, thoughts while walking through a Japanese garden may make one wonder whether one does the thinking spontaneously or if the garden induces the thinking. Creating this type of interaction and conversation between humans and ALife is what we expect from living technology.

Acknowledgments

I thank Yuta Ogai, Motoi Ishibashi, and Mizuki Oka for stimulating discussions and technical support. I thank Kenshu Shintsubo for his camera work for the MTM, and Miyuki Kawamura for organizing the making up of the video. I also thank Evala and Keiichiro Shibuya for helping with the sound effects. I appreciate the Yamaguchi Center for Arts and Media for providing us this opportunity to present the MTM. This work is partially supported by a JSPS Grant-in-Aid for Scientific Research (B) ("Towards the Construction of Technological Philosophical Extension of Ecological Phenomenology") and by a JSPS Grant-in-Aid for Challenging Exploratory Research ("Development of the Revolutionary Experiment Setups for Studying Artificial Life Systems").

References

1. Aihara, K., Takabe, T., & Toyoda, M. (1985). Chaotic neural networks. *Physics Letters A*, *144*(6–7), 333–340.
2. Buckner, R. L., Andrews-Hanna, J. R., & Schacter, D. L. (2008). The brain's default network anatomy, function, and relevance to disease. *Annals of the New York Academy of Sciences*, *1124*, 1–38.
3. Daoyun, J., & Wilson, M. A. (2007). Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature Neuroscience*, *10*(1), 100–107.
4. Hanczyc, M. M., & Ikegami, T. (2010). Chemical basis for minimal cognition. *Artificial Life*, *16*(3), 233–243.
5. Hanczyc, M., Toyota, T., Ikegami, T., Packard, N., & Sugawara, T. (2007). Chemistry at the oil-water interface: Self-propelled oil droplets. *Journal of the American Chemical Society*, *129*(30), 9386–9391.
6. Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley.
7. Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the U.S.A.*, *79*, 2554–2558.
8. Hopfield, J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the U.S.A.*, *81*, 3088–3092.
9. Hubert, J., Matsuda, E., Silverman, E., & Ikegami, T. (2009). A robotic approach to understanding robustness. In *Proceedings of the 3rd Mobiligence Conference* (pp. 361–366).
10. Ikeda, K., Otsuka, K., & Matsumoto, K. (1989). Maxwell-Bloch turbulence. *Progress in Theoretical Physics* (Supplement), *99*, 295–324.
11. Ikegami, T. (2007). Simulating active perception and mental imagery with embodied chaotic itinerancy. *Journal of Consciousness Studies*, *14*, 111–125.
12. Ikegami, T. (2009). Rehabilitating biology as a natural history. *Adaptive Behavior*, *17*, 325–328.
13. Ikegami, T. (2010). *Sustainable autonomy and designing mind time*. ACM Digital Library.
14. Ikegami, T., & Morimoto, G. (2003). Chaotic itinerancy in coupled dynamical recognizers. *Chaos*, *13*, 1133–1147.
15. Ikegami, T., & Suzuki, K. (2008). From homeostatic to homeodynamic self. *BioSystems*, *91*, 388–400.
16. Jen, E. (2005). Stable or robust? What's the difference? In *Robust design: A repertoire of biological, ecological and engineering case studies*. Oxford, UK: Oxford University Press.
17. Kaneko, K., & Tsuda, I. (2003). Chaotic itinerancy. *Chaos*, *13*(3), 926–936.
18. Kitano, H. (2007). Towards a theory of biological robustness. *Molecular Systems Biology*, *137*, 1–7.
19. Libet, B. (2004). *Mind time*. Cambridge, MA: Harvard University Press.
20. Nozawa, H. (1992). A neural network model as a globally coupled map and applications based on chaos. *Chaos*, *2*, 377–386.
21. Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*(3), 267–273.

22. Oka, M., & Ikegami, T. (2013). Exploring default mode and information flow on the web. *PLoS ONE*, *8*(4), e60398.

23. Oudeyer, P.-Y., & Kaplan, F. (2007). What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics*, *1*, 1–14.

24. Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., & Shulman, G. L. (2001). A default mode of brain function. *Proceedings of the National Academy of Sciences of the U.S.A.*, *98*(2), 676–682.

25. Raichle, M. E., & Snyder, A. Z. (2007). A default mode of brain function: A brief history of an evolving idea. *NeuroImage*, *37*(4), 1083–1090.

26. Scheutz, M. (2011). Evolution of affect and communication. In D. Gokcay & G. Yildirim (Eds.), *Affective computing and interaction: Psychological, cognitive and neuroscientific perspectives* (pp. 75–93). Hershey, PA: IGI Global.

27. Silverman, E., & Ikegami, T. (2010). Robustness in artificial life. *International Journal of Bio-Inspired Computation*, *2*(3), 197–203.

28. Van Horn, S. C., Erisir, A., & Sherman, M. S. (2000). Relative distribution of synapses in the A-laminae of the lateral geniculate nucleus of the cat. *Journal of Comparative Neurology*, *416*(4), 509–520.