Progress Report: Active, Real-time Prototype Recognition in a Conceptual Network

Bowen Xu

Department of Computer and Information Sciences,
Temple University, Philadelphia, USA
bowen.xu@temple.edu

May 2, 2024

1 Introduction

Perception is one of the primary and critical aspects of intelligence – it is right above the interface between the mind and the world, and plenty of intelligence phenomena are highly related to perception. Perception is also one of the hardest challenges in AI – given a large number of input signals, how to organize them efficiently and how abstract concepts emerge remain to be answered; although over the years *deep learning* has gained huge success in many domains or problems (especially in *computer vision*), the two issues, the lack of interpretability and the weird behaviors (sometimes called *hallucination* nowadays) by subtle perturbations[1], are apparently two "clouds" upon the deep-learning horizon.

How does humans' perception work? Is there a unified and consistent way, across various sensory channels (including sight/*vision*, hearing/*audition*, taste/*gustation*, smell/*olfaction*, touch/*somatosensation*, *etc.*), of the mind to perceive "open environments"?

There are some direct observations on the process of perceiving environments. For example, humans perceive a scene by *eye movement* (see Fig. 1), and the related research field in psychology is *visual attention* [6]. Humans' perception forms some stable patterns and concepts that are robust to perturbations, and the recognized concepts are invariant to distortion, translation, scale, and even rotation. Besides, the perception procedure is also subjective and highly related to a subject's demand – people tend to recognize what they expect but not everything in a certain context.

Just like the motivations for studying intelligence, the motivations for studying perception are also twofold – (1) to understand human perception procedure

[1] A widely known example is that a deep neural network may recognize a panda as an ostrich when adding some noises to the input image, but the noises do not influence humans at all. Another example is that, after shuffling the positions of the features on a face, humans can easily see the difference, but a convolutional neural network still sees it as a face confidently.

[3] The pictures of the famous painting *Unexpected Visitors* with eye trajectories stem from https://www.cabinetmagazine.org/issues/30/archibald.php.

Fig. 1. Task-oriented Top-Down Attention. The eye trajectories vary in different task-hints [6].[3]

and discover general principles of perception, and (2) to enable machines to perceive the world. "Deep neural networks", due to the lack of interpretability, is not a promising way to achieve the first goal of the research, thus, I suggest following another route, called "conceptual networks", to model perception. More specifically, as suggested by [5], perception is ***subjective***, ***active***, and ***unified***, and the model of perception that I would like to research also has these features – (1) *Subjectivity*: new patterns (also called *schema* [2] or concepts) come into being based on old patterns. This procedure is also called *bootstrapping* [**?**] in cognitive science. (2) *Proactivity*: it should contain a sensory step and a motor step, forming a sensorimotor loop. (3) *Unity*: it should follow a normative theory and a unified representation; specifically, in this research, *Non-Axiomatic Logic* (NAL) [4] is adopted as the theoretical foundation to build the model. To emphasize these features of perception, I suggest using the term *sensorimotor model* to refer to the perceptual model to be researched.

This paper is a progress report on modeling the sensorimotor procedure. Firstly, in Sec. 2, a theory is provided, specifying the overall considerations. Secondly, in Sec. 3, the theory is formalized by extending Non-Axiomatic Logic and designing the control mechanism specific to the sensorimotor procedure. Finally, the current implementation progress and tests are reported in Sec. 4 and Sec. 4.

2 Theory

In perception, on the one hand, *an intelligent system tries to explain and predict sensations via the existing concepts in its memory*; on the other hand, *the system tries to change its perceptive field via actions, for the sake of confirming its explanations and predictions, as well as achieving its desires*. This perspective shares a similar intuition with *Active Inference* in Karl Friston's theory [3], as well as *Assimilation* in Jean Piaget's theory [2]. In the meanwhile, *the system changes its memory to explain past experiences*, and this procedure corresponds to *Accommodation* in Piaget's theory.

In contrast to algorithms in *computer vision*, in this research, the system should not be viewed as an algorithm when perceiving its environment, because the system executes working cycles without an explicit end-point and does not follow the same routes within a problem-solving period (as similar to Non-Axiomatic Reasoning System, NARS [4]). In each *working cycle*, the system accepts sensory input, processing it within a relatively constant time, and decides where to see next, subsequently executing an action to shift its perceptive field. Different from NARS [1] at the current stage, it is assumed that an exhaustive update is acceptable if each step is local and potential specific hardware can process it within a small constant time. For example, in the human brain, each neuron's *membrane voltage* decays in parallel without depending on other neurons, and similar processing is available in neuromorphic hardware. Similarly, if a conceptual network can be implemented in specific hardware, such exhaustive treatment should be legal in theory.

Within each working cycle, sensory signals are discretized and transformed into concepts, so that they can be further handled by the system. The system forms and organizes a conceptual network according to its experience, modifying both the structure and the values in connections. Some concepts are significantly active in a certain context as if the system perceives something representing the outside *objects*.

There are two types of concepts in perception (as shown in Fig. 3):

- A ***composition*** is a special concept composed of a *prototype* as *part* and a *prototype* as *whole*. It represents the relation between *part* and *whole*. A composition is attached with an attribute, *relative location*, which indicates the location of a *part* relative to its *whole* – It is *relative* in the sense that the displacement between two locations can be computed without defining an absolute, original point.

As the implications of this theory, the system would perform some human-like properties – Since the system works with an endless loop, there is no "final report" on "an image's categorization(s)"[4]. By contrast, the system continuously perceives its sensations, so that different concepts may catch its attention at different moments. The system may gradually get a better and better understanding of scenery as time goes by, but it probably loses many details when it is in a hurry.

There are some other notions widely used in *computer vision*, though some of them have quite different interpretations in this work.

- ***Feature*** is the alias of *prototype*.
- An ***object*** is an instance of a prototype. In this sense, *object* here does not mean "a thing as it is", but rather the summary of *relations* among *prototypes*.
- ***Recognition*** is the process in a system to retrieve its memory (*i.e.*, the conceptual network) and pay attention to active concepts.

[4] Nonetheless, in some special cases (*e.g.*, exams), the system can provide its final answer via a decision-making procedure.

3 Model

In this section, I try to formalize the theory proposed above. To better describe perception phenomena, I extend Non-Axiomatic Logic in Sec. 3.1, introducing a new copula "$\mapsto$" to represent the "part-whole" relation, as well as some inference rules for retrieving an object in the memory. The following subsections, Sec. 3.2 and Sec. 3.2, describe the control mechanism of the sensorimotor system. The idealized situation, that specifies the purpose of the system, is provided in Sec. 3.4, and it can be viewed as the rationale of the control mechanism.

3.1 Representation & Inference

A common relation in perception is "part-whole", or *composition*. For example, a bicycle is composed of two wheels and a frame. *Composition* has a different meaning from *intersection* or *union* in current NAL. Concept *bicycle* is not the *intersection* or *union* of concepts *wheel* and *frame*. However, concept *bicycle* is the *extensional intersection* of concepts *vehicle* and *machine*, and concept *bicycle* is also the *intentional intersection* of concepts *human-powered-vehicle* and *two-wheeled-vehicle*. Wheel is not a type of bicycle, thus we cannot use *inheritance* to represent the relation.

For this purpose, another copula is introduced,

Definition 1 *If P and W are events, composition statement "$P \mapsto W$" is true if and only if P's occurrence provides a piece of positive evidence for W's occurrence. The first term P is called part, and the second term W is called whole.*

Intuitively, "$\mapsto$" can be read as "a part of". Usually, location of *part* matters in perception. Similar to the treatment of *time* in NAL-6, *location* is attached to a *composition statement*,

Definition 2 *"$(P_1 \mapsto W)[l_1] \bigwedge (P_2 \mapsto W)[l_2]$" is true if and only if "$((P_1, \Uparrow move(l_2 - l_1), P_2) \leftrightarrow W) \bigwedge ((P_2, \Uparrow move(l_1 - l_2), P_1) \leftrightarrow W)$", where "$\Uparrow move(\#1)$" is a mental operation that shifts the system's perceptual focus by distance $\#1$.*

Locations here are *relative* to *whole* term W. It implies that an absolute origin of a coordinate system is not needed.

Definition 3 *A location attached to a prototype, e.g., "$P[l]$", is implicitly relative to a whole term. What the whole term is depends on context.*

Definition 4 *Given composition statements "$(P_1 \mapsto W)[l_1], ..., (P_2 \mapsto W)[l_n]$", the whole term W is also represented as a compound term "$W : \{|C_1[l_1], ..., C_n[l_n]|\}$" or anonymously "$\{|C_1[l_1], ..., C_n[l_n]|\}$".*

A *whole* can be represented as a compound of its *parts*, and Def. 4 provides the formal representation. A *whole* term identifies a *prototype* defined in Sec. 2.

For inference, the *part-whole rule* has the following form:

$$\{(P \mapsto W)[l_1]\langle f_1, c_1\rangle, P[l_2]\langle f_2, c_2\rangle\} \vdash W[l_2]\langle F_{spj}F_{prt}\rangle$$

where the $F_{spj}F_{prt}$ is the combination of two truth-functions,

$$\begin{aligned} f, c' =& F_{prt}(f_1, f_2, c_1, c_2) \\ c =& F_{spj}(c', l_1, l_2) \end{aligned}$$

In the first step, *frequency* and *confidence* are computed by the "part-whole" function F_{prt}, and then the *confidence* is decayed via the *spatial-projection* function F_{spj}. The definitions of F_{prt} and F_{spj} are shown in Tab. 1. According to Def. 1, the total evidence is no more than 1, and the *frequency* depends on that of the two premises. The *confidence* is decreased in *spatial projection*, and the extent depends on the distance between the two locations. In this paper, a bell-shaped function is adopted (see Fig. 2) for *spatial projection*. Here, l_1 and l_2 are relative to W.

The *revision rule* with spatial information has the following form:

$$\{W_1[l_1]\langle f_1, c_1\rangle, W[l_2]\langle f_2, c_2\rangle\} \vdash W[l]\langle F_{srv}F_{rev}\rangle$$

where the $F_{srv}F_{rev}$ is the combination of two truth-functions,

$$\begin{aligned} f, c =& F_{rev}(f_1, f_2, c_1, c_2) \\ l =& F_{srv}(f_1, f_2, c_1, c_2, l_1, l_2) \end{aligned}$$

In the first step, *frequency* and *confidence* are computed by the *revision* function F_{rev} [4], and then the *location* is adjusted via the *spatial-revision* function F_{srv}. The definition of F_{srv} is shown in Tab. 1. Here, l_1, l_2, and l are relative to W.

Table 1. The Truth-Value Functions of Inference Rules

type	**inference**	**name**	**function**
spatial revision	spatial revision	F_{srv}	$l = \frac{l_1 f_1 c_1 + l_2 f_2 c_2}{f_1 c_1 + f_2 c_2 + \epsilon}$
immediate inference	spatial projection	F_{spj}	$c = bell(l_1 - l_2) \times c'$
weak syllogism	part-whole	F_{prt}	$f = and(f_1, f_2)$ $w = and(f_1, f_2, c_1, c_2)$

3.2 Memory

For the moment, the memory is a single-layered conceptual network (see Fig. 3). An input *feature* corresponds to a *whole* term that names a concept in

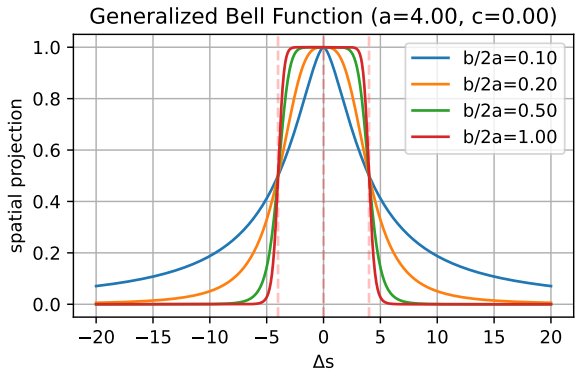

Fig. 2. Bell-shaped Function for Spatial Projection: $y = \frac{1}{1 + \left|\frac{\Delta s - c}{a}\right|^{2b}}$

the memory. A *whole* term is regarded as a *part* term of a whole at the higher level. A *whole* term in the higher level is a *prototype* of its *parts*. A connection from a *part* term to a *whole* term corresponds to a *composition statement*.

Each concept or connection consists of some truth-values and budgets:

- $truth_e$: the truth-value indicating the occurrence of an event.
- $truth_a$: the truth-value indicating the anticipation of an event.
- $budget_c$: the budget for the competition of temporal resources (*i.e.*, processing time).
- $budget_m$: the budget for the competition of spatial resources (*i.e.*, storage or memory).

The difference between $truth_e$ and $truth_a$ is that the former is increased when an event truly occurs, while the latter is modified if an event is anticipated to occur. Truth-value of anticipation is necessary because anticipation plays a role in accumulating negative evidence – a human usually does not think about a non-occurring object, unless it conflicts with his anticipation(s), as implied by many works in psychology.

The budgets for temporal and spatial resources are separated (as different from NARS [4,1]) for some reasons: 1) in some cases, an item, that should be remembered in a short period, should be forgotten in the long period; thus *priority* and *durability* of $budget_m$ serves for the balance between long-term and short-term storage. If a large number of items swarm into the memory, *durability* is critical to maintain those valuable items for the long term, since it avoids having them squeezed out of memory. 2) In some cases, some long-standing items should be recalled and processed at times if there is nothing urgent at hand. Thus, there should be a balance between long-term processing and short-term processing. *Priority* and *durability* in $budget_c$ help to balance them. 3) In some cases, an item, that should be remembered in a short period, does not need to be processed urgently, and a single *priority* failed to resolve this conflict.

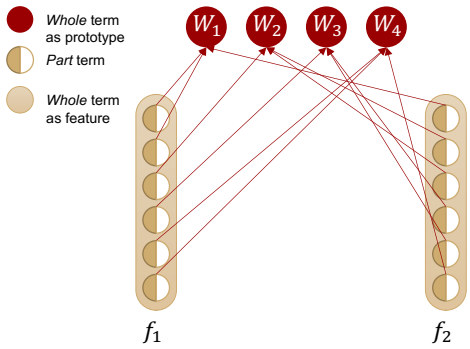

Fig. 3. The Structure of the Conceptual Network

In this work, only one type of relation among concepts is considered, that is, the "part-whole" relation "$\mapsto$" – a component is a part of a prototype to some degree. The truth-value (denoted as $truth_p$) of a *composition statement* is eternal, meaning that it does not decay through time. In contrast, the truth-value of event-occurrence ($truth_e$) and that of anticipation ($truth_a$) are time-sensitive.

3.3 Working Cycle

Given the structure, the tricky part is how to exploit it and modify it. Similar to NARS, the system in this paper also conducts working cycles repeatedly. In each working cycle, the system retrieves some concepts, builds or deletes connections among them, and revises the truth-values of connections.

More specifically, it involves four aspects,

1. **Retrieving**, which is majorly related to manipulating *budgets* and *anticipations* of concepts,
2. **Hypothesizing**, *i.e.*, constructing new concepts and "part-while" relations,
3. **Revising**, *i.e.*, accumulating evidence for truth-values, and
4. **Recycling**, *i.e.*, removing concepts and relations because of insufficient resources.

The current design only involves the *retrieving* aspect, while leaving the other aspects to future work.

In retrieving, due to the relativity of space, when a feature occurs, the system sometimes cannot determine what *composition* it pertains to. For example, a prototype W is composed of three *parts* at different relative locations, *i.e.*,"$W : \{|P_1[(0.0; 0.0)], P_1[(0.2, 0.0)], P_1[(0.1, 0.2)]|\}$".[5] When feature P_1 occurs, "$P_1[(0.0; 0.0)]$", "$P_1[(0.2, 0.0)]$", "$P_1[(0.1, 0.2)]$" are all possible *part* terms. The

[5] Note that *location* here is represented as a two-dimensional number, but only the difference between two locations makes sense. For example, this prototype is equivalent to "$\{|P1[(0.1; 0.1)], P1[(0.3, 0.1)], P1[(0.2, 0.3)]|\}$".

system generates six potential actions, "$\Uparrow move((0.2, 0.0))$", "$\Uparrow move((0.1, 0.2))$", "$\Uparrow move((-0.2, 0.0))$", "$\Uparrow move((-0.1, 0.2))$", "$\Uparrow move((-0.1, -0.2))$", and "$\Uparrow move((0.1, -0.2))$". After moving, some more actions will be generated. This procedure is similar to search in computer science, and actions lead to different states (*i.e.*, budgets and truth-values) of the system. In contrast to the traditional search algorithms, in the *retrieving* procedure, we do not assume a *global* description of the system or the environment, and there is no explicit end of the procedure. The system continuously looks for concepts that better explain current situations.

To record the intermediate states, a data-structure *task* is adopted. A *task* consists of a location l_{task} relative to the prototype, and *mirrors* of *compositions*. A *mirror* of composition has the same meaning as the corresponding composition, except that a copy of the budgets and the anticipated truth-value is maintained in the *mirror*. If the system focuses on merely one single task, it goes through all the compositions and matches them with the prototype that the *task* corresponds to, modifying the truth-value of the prototype's occurrence. Usually, there is more than one task, each of which corresponds to one possible explanation of the input signals. When the system meets more and more mismatching in a *task*, the *priority* in the *task*'s $budget_c$ becomes lower and lower, so that other *tasks* get a higher chance to be concerned.

In a nutshell, in each working cycle, the system repeatedly does the same things, 1) to accept one event and conceptualize it, 2) to pick out one prototype with the maximal priority from the memory, 3) to pick out one task with the maximal priority from the prototype, 4) to pick up one (mirror of) composition with the maximal priority from the task, 5) to predict the occurrence of another composition and take an action if necessary, and 6) (not designed yet,) to modify the memory.

3.4 The Idealized Situation

Each working cycle can be viewed as a step of an optimization procedure, where the purpose is to find some concepts that best explain the input experience, such that the following loss J_{explain} is minimized:

$$J_{\text{explain}}(\mathcal{I}, \mathcal{C}) = \alpha \cdot match(\mathcal{I}, \mathcal{C}) + \beta \cdot complexity(\mathcal{C}) + \gamma \cdot utility(\mathcal{C}) \quad (1)$$

where $\mathcal{C}$ is the set of concepts used to explain the image, and $\mathcal{I}$ is the input features in the input stimuli. Function $match(\mathcal{I}, \mathcal{C})$ evaluates how well the concepts $\mathcal{C}$ matches the input stimuli $\mathcal{I}$.

To recognize *objects* is to find a set of concepts $\mathcal{C}$ that minimize $J_{explain}$:

$$\mathcal{C} = \arg\min_{\mathcal{C}_i} J_{\text{explain}}(s, \mathcal{C}_i) \quad (2)$$

An agent could traverse all possibilities exhaustively if having sufficient computing resources. However, an agent like human should probably do recognition more smartly, basically due to AIKR – the system does not afford such a large overhead given insufficient computing resources.

4 Implementation Progress & Working Examples

Ideally, the system should learn some prototypes or patterns from noisy samples in real-time, and it should be able to recognize the categories from tens of thousands or even millions of prototypes. This is the actual case faced by the human mind. However, in the initial stage, due to insufficient resources and knowledge, I was unable to achieve it in one fell swoop. In this paper, I consider merely the simplest example – three features (of the same type) with different locations that form a triangle (see Fig. 4) – and the learning process (hypothesizing, revising, and retrieving mentioned in Sec. 3.3) is not designed and implemented yet. The system is able to recognize a prototype from samples.

Even for implementing the retrieving process, I split it into two sub-steps. In the first step, I implemented an algorithm that matches a given prototype in real-time (see Algorithm 1). As shown in Fig. 5, the system focuses on different parts, and the "matching value" increases to a relatively high value, meaning that the prototype is recognized. In the second step, I transfer the algorithm into the working cycle (see Algorithm 2). The tricky part is adjusting the budgets, so as to direct the system's attention. This part is not well elaborated in Sec. 3, because there are some issues unsolved and it is not mature enough. As shown in Fig. 6, in most cases, the system converges to the correct prototype with the correct location, however, sometimes the system sticks into a wrong obsessiveness and cannot correct itself.

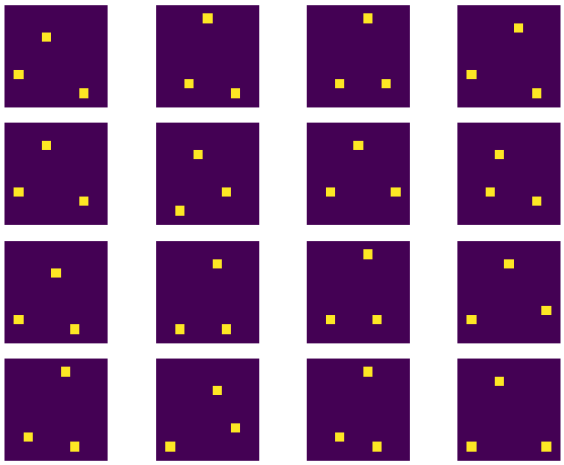

Fig. 4. Input Examples

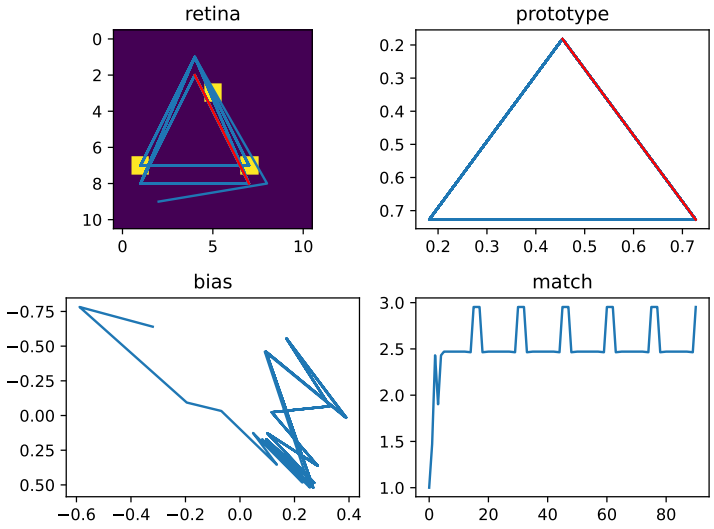

Fig. 5. The matching algorithm (to watch the video attached, see Appendix A)

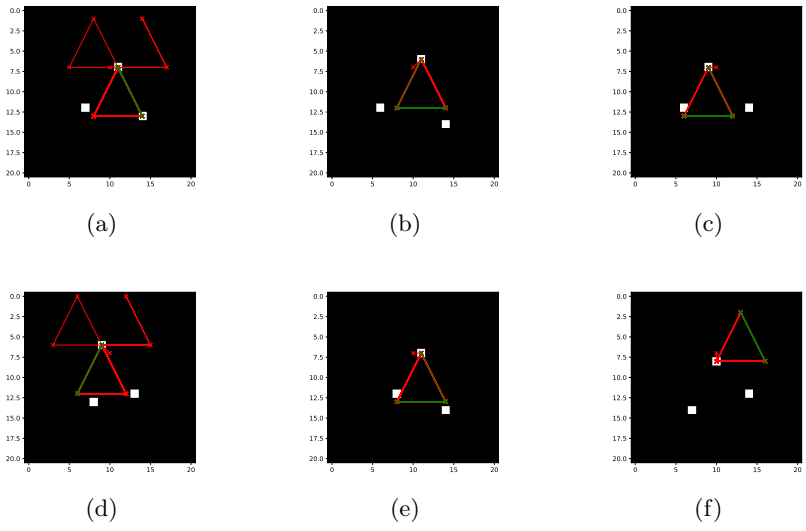

Fig. 6. (a)∼(d) success examples; (f) a failure example

5 Acknowledgment

I would like to express my gratitude to Pei Wang for discussing with me and sharing his ideas. Additionally, I thank Zhengyu Liu for the insightful discussions we shared.

References

1. Hammer, P., Lofthouse, T., Wang, P.: The opennars implementation of the non-axiomatic reasoning system. In: Artificial General Intelligence: 9th International Conference, AGI 2016, New York, NY, USA, July 16-19, 2016, Proceedings 9. pp. 160–170. Springer (2016)
2. Müller, U., Ten Eycke, K., Baker, L.: Piaget's Theory of Intelligence. In: Goldstein, S., Princiotta, D., Naglieri, J.A. (eds.) Handbook of Intelligence: Evolutionary Theory, Historical Perspective, and Current Concepts, pp. 137–151. Springer, New York, NY (2015). https://doi.org/10.1007/978-1-4939-1562-$0_1$0
3. Parr, T., Pezzulo, G., Friston, K.J.: Active Inference: The Free Energy Principle in Mind, Brain, and Behavior. The MIT Press (Mar 2022). https://doi.org/10.7551/mitpress/12441.001.0001
4. Wang, P.: Non-axiomatic logic: A model of intelligent reasoning. World Scientific (2013)
5. Wang, P., Hahm, C., Hammer, P.: A Model of Unified Perception and Cognition. Frontiers in Artificial Intelligence **5** (2022), https://www.frontiersin.org/articles/10.3389/frai.2022.806403
6. Yarbus, A.L.: Eye Movements and Vision. Springer US, Boston, MA (1967). https://doi.org/10.1007/978-1-4899-5379-7, http://link.springer.com/10.1007/978-1-4899-5379-7

A Supplementary Materials

The source code is attached. See the file "readme.txt". A video "demo1.mov" demonstrating Algorithm 1 is also attached.

B Algorithms

Algorithm 1 Brute-Force-Matching(p, c_0, R, k)
// p: prototype
// c_0: component as the start-point
// R: retina object
// k: the number of iterations

```
1  visited = dict({})
2  value = 1.0
3  c_0.budget_c.inhibit(0.1)
4  visited[c_0] = value
5  bias = (0.0, 0.0)
6  for i = 1 to k × p.length
7      c = pick out the item with the maximal priority of budget_c in p
8      loc_0 = get the relative location of c_0
9      loc_c = get the relative location of c
10     mv_1 = (loc_c − loc_0) × p.scale
11     loc_R = R.loc
12     mv_2 = R.move(mv_1)
13     err = mv_1 − mv_1 + bias × p.scale
14     patch = R.sense(err)
15     feat, bias_f = get the feature closest to R.loc + err and the location bias
16     if feat ≠ NULL
17         v = evaluate the value given bias_f
18         if c in visited
19             value = value − visited[c_0]
20         visited[c] = v
21         bias = bias × value + bias_f × v
22         value = value + v
23         c_0 = c
24     else
25         R.move_to(loc_R)
26     c.budget_c.inhibit(0.1)
27 return value
```

Algorithm 2 Real-Time-Retrieving(*sml*, R)
// *sml*: sensorimotor layer
// R: retina object

1 $s = R.sense()$
2 Match existing prototypes given s
3 Modify the priority values of $cpnt.budget_c$, $inst.budget_c$ and $proto.budget_c$
4 $proto$ = pick out a prototype with maximal $budget_m.priority$ from *sml*
5 $task$ = pick out an task with maximal $budget_m.priority$ from $proto$
6 $cpnt$ = pick out an component with maximal $budget_m.priority$ from $task$
7 Decrease the priority value of $cpnt.budget_c$
8 $cpnt_{next}$ = pick out an component with maximal $budget_m.priority$ from $inst$
9 Anticipate $cpnt_{next}$ (*i.e.*, increase $truth_a$)
10 $\Delta l = cpnt_{next}.location$ - $inst.location$
11 $\Delta l' = R.move(\Delta l)$
12 $sml.motor\_input(\Delta l')$