The construction and acquisition of visual categories

Tony Belpaeme[1], Luc Steels[1,2], and Joris Van Looveren[1]

[1] Vrije Universiteit Brussel
Artificial Intelligence Laboratory
Pleinlaan 2, 1050 Brussels, Belgium
steels@arti.vub.ac.be
http://arti.vub.ac.be
[2] Sony Computer Science Laboratory Paris

Abstract. The paper proposes a selectionist architecture for the construction and acquisition of visual categories by a robotic agent. The components of this architecture are described in some detail and results from experiments on physical robots are reported.

Keywords: origins of language, self-organization, distributed agents, open systems, vision.

1 Introduction

A cognitive agent must have a way to map symbolic categories onto its experiential, perceptually acquired data streams. For example, in a task requiring language communication or symbolic planning, the data gathered from visual input through a camera must somehow be categorised and interpreted. The problem of bridging the gap between the subsymbolic world of continuous dynamics and real world sensing and effecting and the symbolic world of categories and language, has turned out to be very difficult. No successful general purpose vision system has emerged that is capable to perform the mapping task in real-time. Part of this is due to the complexity of the task, requiring vast amounts of computation power and memory. It is also due to the enormous richness of real-world data and the difficulty to extract regularity.

So far two approaches dominate the field. The first approach, pioneered and exemplified by the work of David Marr [5] consists in programming explicitly complex feature detectors which operate on successive stages of information processing. The end result is a symbolic 3-d model of the scene. Although a large amount of practically usable image processing components have resulted from this approach, the ultimate goal of a general purpose vision system that could interface with a symbolic knowledge representation component has not materialised. Some researchers who have at one point been fully involved with this approach, notably Brooks [14], have decided to give up altogether and insist on staying at the subsymbolic level, at least for the control of behaviors. Others

such as Ullman [15] have introduced more focus by limiting the scope of vision to special-purpose detection handled by so called visual routines.

A second approach is associated with neural network modeling. It relies on learning mechanisms to acquire the feature detectors by continuous exposure to a series of examples and counter examples. This approach has also a long tradition, starting with the perceptron, and advancing with the discovery of the back-propagation learning algorithm, Kohonen's self-organising feature maps, etc. In general, the neural network approach has been able to come up with solutions that are robust against the natural variation occuring in real-world data. But there are two difficulties. The supervised learning systems rely on an outside intelligence to come up with the initial series of examples. Such a setup is not possible for autonomous robotic agents which may find themselves in environments that are unknown to humans or perceivable through sensory modalities (such as infrared) to which humans have no access. The unsupervised learning systems on the other hand are only constrained by the regularities in the data themselves. They are inductive mechanisms that cannot be steered to develop categories under the influence of certain tasks.

This paper reports on experiments which explore an alternative to the approaches briefly introduced above. Our first hypothesis is that the vision module should be made responsible for less. We make a distinction between three components:

1. The *vision component* which is responsible for segmenting and data collection.
2. The *categorisation component* which performs the transition from real world data to symbolic descriptions.
3. The *user component* which is another process that needs the symbolic descriptions in one way or another. This could be a language task or a planning task.

Depending on the task (i.e. the user component) the categorisation may be different. So no general purpose visual categorisation is sought for, on the contrary. The categories should be adapted to the task.

Our second hypothesis is that each component is constructed and adapted through a selectionist process. A selectionist process contains on the one hand a generator of diversity which is not driven by the task at hand, and on the other hand a process maintaining or eliminating variations based on their performance, i.e. how well they do with respect to a set of selectionist pressures provided by the context or users of the result. Evolution of species by natural selection, as proposed by Darwin, or reenforcement of natural variation in the immune system, as proposed by Jerne and Edelman, are two examples of selectionist processes. But the idea can be applied to any system that can be made to exhibit spontaneous variation and subsequent selection.

This paper focuses only on how the categorisation component could self-organise. We assume that category buildup proceeds by the spontaneous creation of new distinctions when the user component provides feedback that the existing set of categories are insufficient. Whether the new distinction is adequate will

depend on later evaluations. The selectionist pressure in other words comes from the user component. The same scheme could be applied to the vision component. In particular, the vision component could internally have a generator of diversity that spontaneously creates new data if the categorisation module provides feedback that the existing data is not sufficient to construct adequate categorisations. The categorisation module acts in this case as the source of selectionist pressure.

To make the paper concrete, we assume that the user component is a language system that is itself also selectionist. The language system lexicalises categories or combinations of categories called feature structures. Selectionist pressure now comes from whether a particular lexicalisation gains acceptance in the rest of the community. In this paper, the "talking heads" experiment is taken as a source of examples. In this experiments two robotic heads which can track moving objects based on visual input, watch a static or dynamic scene. The robots must develop from scratch visual categories which are adequate for describing what they see. The language itself also develops from scratch and without human intervention.

Other papers have provided details on the language component and how this component interacts with a (selectionist) categorisation module (see e.g. [8] or [10]). This paper focuses on the interaction between the vision component and the categorisation component. The vision component itself is assumed to provide a fixed set of data. The present research has first been validated with software simulations and then ported to real world physical robots (see also [13]).

The rest of the paper is in four parts. The next part describes the robotic set-up used in the present experiments. The third part describes the vision component as it is currently implemented. The fourth part describes the categorisation component. Then the results of some experiments are presented in some detail.

2 The Talking Heads Experiment

A *talking head* consists of a black and white camera mounted on a pan-tilt unit, electronics to perform low-level signal processing and actuator control, and two PCs performing visual processing and symbol processing respectively. The experiment uses two heads. Both stand next to each other, observing the same scene. This scene contains several objects in various shapes and sizes. The scene can even contain moving objects, such as a robot driving around and pushing other objects.

The visual processing filters the incoming image from the head. In the next step, some visual modules are unleashed on the filtered image. A module is specifically for a certain feature of the image, e.g. a specific color. In the current implementation three modules are used: one module is active for motion in the image, one for detecting patches of a light intensity and one for detecting patches with a dark intensity. The module detecting motion in the image is also connected to the control of the pan-tilt motors of the head. This results in the heads focussing on motion in the scene.

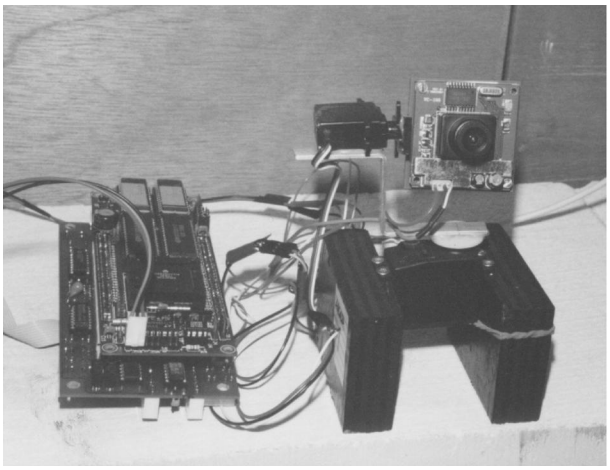

Fig. 1. The eye of a "talking head" with associated electronics for low level signal processing and real-time actuator control needed for tracking.

Each visual module singles out patches in the image which are aggregated to segments withing a rectangular bounding box. Overlapping segments are joined and segments that are too small for further analysis neglected. In a next step, data for each patch (such as average intensity, position in the image, ...) is calculated. All these patches and their features are continuously being logged, along with the first derivative in time of each feature (as to detect changes over time) and the time at which a patch appeared or disappeared in the image.

After a few moments of observing the scene, the heads agree to communicate. At this moment, all information on the patches in the image logged up until now is passed on to the categorisation and lexicon components. When this is finished, the heads continue observing the scene.

3 The visual processing

The visual perception is inspired by active vision (for an overview see [1] or [2]). The early visual processing contains bottom-up, uniform and unarticulated processing of the raw image; such as filtering or transforming the image in a more useful representation. This representation is then fed to the visual modules responsible for producing the data used in the language formation. The three modules used are now described in some detail.

The motion module detects motion in the image. This is done by subtracting two subsequent image frames. $f(x, y, t_j)$ is an image taken at time j, $motion(x, y)$ is an array which will contain a 1 if pixel (x, y) moved, or a 0 if not. The threshold $\vartheta > 0$ is used to filter noise.

$$motion(x, y) = \begin{cases} 1 \text{ if } |f(x, y, t_{i+1}) - f(x, y, t_i)| > \vartheta \\ 0 \text{ otherwise} \end{cases}$$

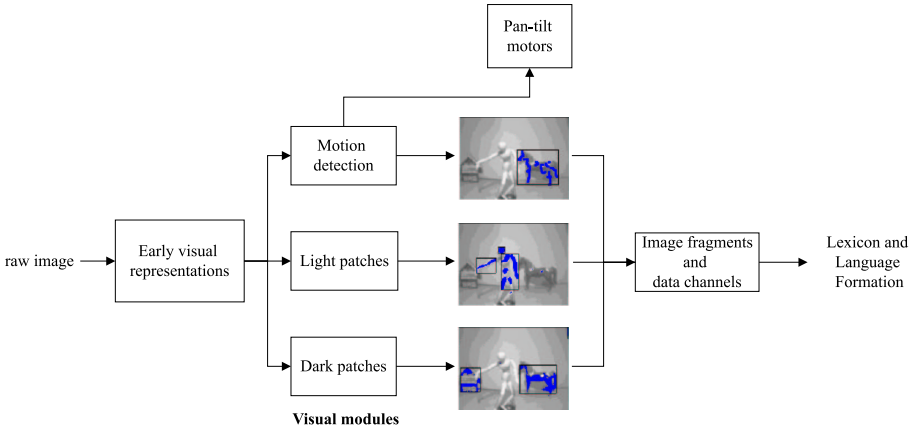

Fig. 2. schematic overview of the visual processing for one talking head.

A bounding box is placed around a consistent set of moving pixels. The centroid of all moving pixels is also calculated and passed on to the control of the pan and tilt motors of the head. When the centroid tends to move outside the image, the motors of the head are repositioned in order to get the centroid of motion back in the middle of the image. The head will always focus on regions with lots of movement, e.g. a robot driving around or a person walking by. The camera does not move as long as the centroid is near the middle of the image, thus resembling saccadic eye movements observed in mammals. Hence we call this *saccadic tracking*.

The light patches module returns all segments in the image of size $w \times h$ having a high intensity. First the average intensity $\bar{I}$ of all pixels and the standard deviation δ of the intensity are calculated.

$$\bar{I} = E(f(x,y,t_i)) \quad \text{with} \quad E(x) = \frac{\sum x}{\text{w.h}}$$

$$\delta = \sqrt{E(f^2(x,y,t_i)) + (E(f(x,y,t_i)))^2}$$

Next, every pixel is checked for being lighter than $\bar{I} + \delta$. $light(x,y)$ is an array containing 1 at position (x,y) if the pixel is light, and 0 if not.

$$light(x,y) = \begin{cases} 1 \text{ if } f(x,y,t_i) > \bar{I} + \delta \\ 0 \text{ otherwise} \end{cases}$$

The dark patches module is the same as the previous module, but now pixels with an intensity lower than $\bar{I} - \delta$ are considered.

$$dark(x,y) = \begin{cases} 1 \text{ if } f(x,y,t_i) < \bar{I} - \delta \\ 0 \text{ otherwise} \end{cases}$$

Image segments are groups of consistent pixels. The way image segments are made is identical for each visual module. Here the algorithm, which resembles region growing from traditional computer vision, is explaned for the light patches module.

1. First, a non-zero entry (a, b) is picked from the array $light(x, y)$. Add (a, b) to a new, empty image segment.
2. For each pixel in the image segment: if a neighboring pixel is non-zero, add it to the image segment. Delete the entry in array $light(x, y)$.
3. Redo the previous step, until all neighboring pixels are zero.
4. Calculate a bounding box around this image segment. If an image element has less than β pixels (an arbitrary threshold) then it is not used in further processing.
5. Take a new non-zero entry, and start from step 1. Stop if no non-zero entries are left in $light(x, y)$.

It should be noted that no object recognition is involved; the visual processing does not recognize or select objects. Anything in the scene, even the floor or the background, can and will be analyzed if it triggers a visual module.
An image segment is a region in the image where every pixel shares a common property. Now, for every image segment, some *sensory channels* are calculated. Sensory channels are real-valued properties of a restricted region in an image. The following sensory channels are used:

Vertical angle of the image segment, relative to the observer (i.e. the talking head).

Horizontal angle of the image segment.

Average intensity of the image segment.

Area of the image segment. This correlates typically with the distance to the observer. The further an image segment is from the observer, the smaller its area will be.

Visibility is a measure for how close an image segment is to the focus of attention: the closer the segment is to the FOA, the higher the visibility will be. The FOA is determined by the centroid of motion, calculated by the motion module.

Fill ratio is the ratio of the pixels marked as interesting by the visual module and the size of the rectangular bounding box.

In figure 3 the output of the visual processing is shown. The scene is static and contains a house, a wooden puppet and a toy horse. Bounding boxes are placed around interesting image segments. The transcript (see table 1) of the data produced by the visual processing shows each image segment, with the coordinates of its bounding box, the start and stop time in milliseconds at which it was visible and the data calculated for each segment. Next to each data value d, $\delta d/\delta t$ is given; since the scene is static, all $\delta d/\delta t = 0$.

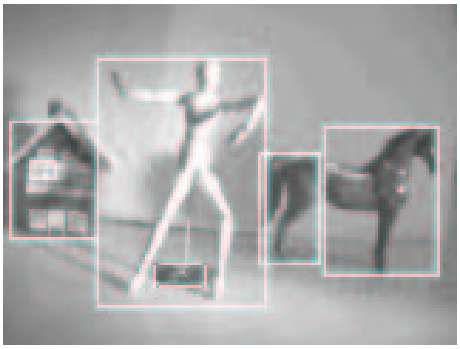

Fig. 3. A screenshot of the output of the visual processing.

4 Ontology creation through discrimination games

We now turn to the categorisation module which starts from the output of the vision module.

Let there be a set of segments $\mathcal{O} = \{o_1, ..., o_m\}$ and a set of sensory channels $S = \{\sigma_1, ..., \sigma_n\}$, being real-valued partial functions over $\mathcal{O}$. Each function σ_j defines a value $0.0 \leq \sigma_j(o_i) \leq 1.0$ for each segment o_i.

An agent a has a set of feature detectors $D_a = \{d_{a,1}, ..., d_{a,m}\}$. A *feature detector* $d_{a,k} = \langle p_{a,k}, V_{a,k}, \phi_{a,k}, \sigma_j \rangle$ has an attribute name $p_{a,k}$, a set of possible values $V_{a,k}$, a partial function $\phi_{a,k}$, and a sensory channel σ_j. The result of applying a feature detector $d_{a,k}$ to an object o_i is a feature written as a pair $(p_{a,k}\ v)$ where p is the attribute name and $v = \phi_{a,k}(\sigma_j(o_i)) \in V_{a,k}$ the value.

The *feature set* of a for o_i is defined as $F_{a,o_i} = \{(p_{a,k}\ v) \mid d_{a,k} \in D_a, d_{a,k} = \langle p_{a,k}, V_{a,k}, \phi_{a,k}, \sigma_j \rangle, v = \phi_{a,k}(\sigma_j(o_i))\}$. Two features $(a_1\ v_1)$, $(a_2\ v_2)$ are *distinctive* iff $a_1 = a_2$ *and* $v_1 \neq v_2$. A distinctive feature set D^C_{a,o_t} is a set of features distinguishing an segment o_t from a set of other segments C. $D^C_{a,o_t} = \{f \mid f = (p\ v) \in F_{a,o_t}$ *and* $\forall o_c \in C$ *either* $/ \exists f' = (p'\ v') \in F_{a,o_c}$ *with* $p = p'$ *or* $\exists f' \in F_{a,o_c}$ *with* f *and* f' *distinctive*$\}$. Clearly there can be several distinctive feature sets for the same o_t and C, or none.

A discrimination game $d =< a, o_t, C >$ involves an agent a, a topic $o_t \in C \subseteq \mathcal{O}$. C is called the context. The outcome of the game is twofold. Either a distinctive feature set could be found, $D^C_{a,o_t} \neq \emptyset$, and the game ends in success, or no such feature set could be found, $D^C_{a,o_t} = \emptyset$, and the game ends in failure.

As part of each game the repertoire of meanings is adjusted in the following way by the agent:

1. $D^C_{a,o_t} = \emptyset$, i.e. the game is unsuccessful. This implies that there are not enough distinctions and therefore $\exists o_c \in C$, $F_{a,o_t} \subseteq F_{a,o_c}$. There are two ways to remedy the situation:

Table 1. Sensory channels corresponding to figure 3. Each line shows the coordinates (left, top, right, bottom) of the bounding box, the start and stop times in milliseconds when the box was seen and the values for the different sensory channels together with their relative change during the time interval. The data channels respectively are: the horizontal and vertical angle, the visibility, the area, the intensity and the fill-ratio. Since this is a still image, all first derivatives in time are equal to zero.

```
((33 19 92 106)   50846 54138
  (0.465441 0.0) (0.503695 0.0) (0.515432 -0.0) (0.852174 -0.0) (0.448984 0.0) (0.388662 0.0))
((9 55 18 61)     50846 54138
  (0.399865 -0.0) (0.493691 0.0) (0.553222 0.0) (0.046518 -0.0) (0.866551 -0.0)(0.529951 0.0))
((2 41 33 82)     50846 54138
  (0.407428 0.0) (0.497312 0.0) (0.547630 -0.0) (0.188681 0.0) (0.284191 0.0) (0.332916 0.0))
((113 43 153 95)  50846 54138
  (0.570797 0.0) (0.522586 0.0) (0.453309 0.0) (0.339950 0.0) (0.145960 0.0) (0.268578 0.0))
((90 52 111 91)   50846 54138
  (0.531648 0.0) (0.528035 -0.0) (0.470159 0.0) (0.138996 0.0) (0.342711 0.0) (0.408776 0.0))
((53 91 71 99)    50846 54138
  (0.473048 0.0) (0.592732 0.0) (0.467110 -0.0) (0.025958 0.0) (0.270678 0.0) (0.542513 0.0))
((53 91 71 99)    50846 54138
  (0.470968 0.0) (0.595915 0.0) (0.466558 0.0) (0.024000 0.0) (0.337255 0.0) (0.569444 0.0))
```

   (a) If there are still sensory channels for which there are no feature detectors, a new feature detector may be constructed. This option is preferred.
   (b) Otherwise, an existing attribute may be refined by creating a new feature detector that further segments the region covered by one of the existing attributes.

2. $D^C_{a,o_t} \neq \emptyset$. In case there is more than one possibility, feature sets are ordered based on preference criteria. The 'best' feature set is chosen and used as outcome of the discrimination game. The record of use of the features which form part of the chosen set is augmented. The criteria are as follows:
   (a) The smallest set is preferred. Thus the least number of features are used.
   (b) In case of equal size, it is the set in which the features imply the smallest number of segmentations. Thus the most abstract features are chosen.
   (c) In case of equal depth of segmentation, it is the set of which the features have been used the most. This ensures that a minimal set of features develops.

The whole system is selectionist. Failure to discriminate creates pressure to create new feature detectors. However the new feature detector is not guaranteed to do the job. It will be tried later and only thrive in the population of feature detectors if it is indeed successful in performing discriminations.

As mentioned earlier, the categorisation component is next coupled to a language component, which in its simplest form constructs a lexicon. The language component provides feedback about which categories have lead to successful language games, and consequently which categories made sense in conversations with other agents. More details about this language component can be found in [10].

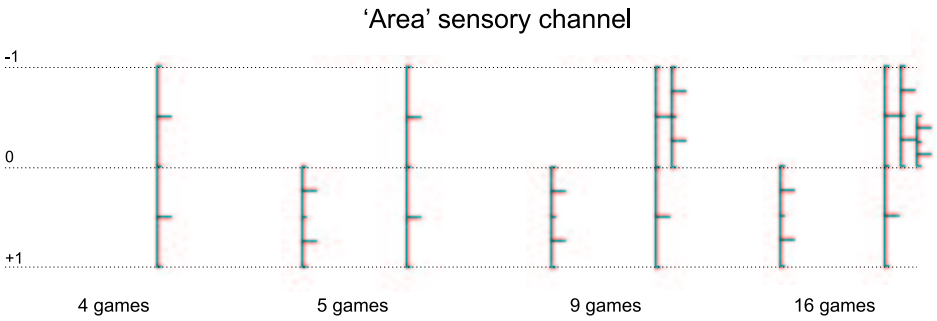

Fig. 4. An example of consecutive refinements made for the 'area' category in one of the agents. The snapshots have been taken after resp. 4, 5, 9 and 16 language games. The left axis ranges from 0 to 1 and represents the average area over a short period of time. The right axis ranges from -1 to 1 and represents the time derivative of the average area during the same period. Note that this examples is not taken from figure 3.

5 Results

Here are some examples of interactions. In the first one the speaker fails to conceptualise the scene and creates a new category by dividing the sensory channel called fill-ratio into two segments associated with the values v-81 and v-82.

```
0. Speaker: Head-16. Hearer: Head-17. Topic: o4.
  Repair Head-16:
   Extend categories: FILL-RATIO [-1.0 1.0]: v-81 v-82
  ? ? => ? ? [failure]
```

In the next game, there is another failure and a new distinction is created now on the visibility channel:

```
1. Speaker: Head-16. Hearer: Head-17.
  Repair Head-16:
   Extend categories: VISIBILITY [-1.0 1.0]: v-83 v-84
  ? ? => ? ? [failure]
```

In game 4, a set of distinctive features has been found but there is no word yet. The speaker creates a new word:

```
4. Speaker: Head-16. Hearer: Head-17. Topic: o12.
  Repair Head-16:
     Extend word repertoire: "(d u)"
     Extend lexicon: ((visibility v-88)) (d u)
  ((visibility v-88)) (d u) => ? ? [failure]
```

In the following game, the speaker is able to find a distinctive feature set and a word, but the hearer is missing the required distinctions:

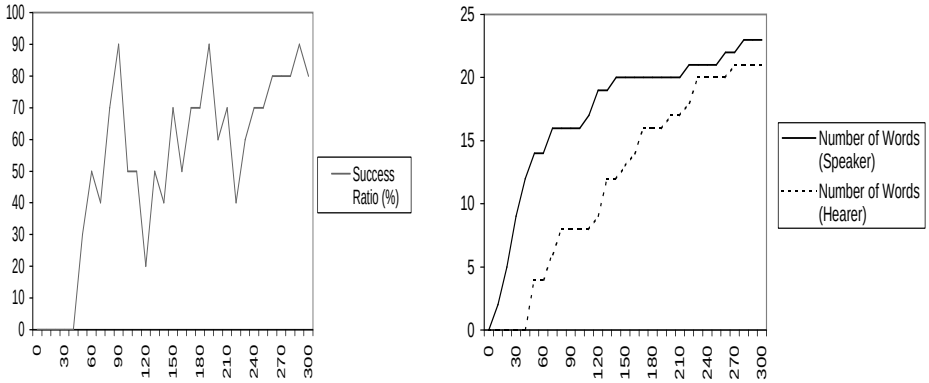

Fig. 5. Graph showing the increase in average communicative success (left) as well as the increase in the number of words in the vocabularies of two robotic heads (right).

```
6. Speaker: Head-16. Hearer: Head-17. Topic: o10.
  Repair Head-17:
    Extend categories: VISIBILITY [-1.0 1.0]: v-91 v-92
  ((visibility v-88)) (d u) => (d u) ? [failure]
```

The first successful game happens after 47 games:

```
47. Speaker: Head-16. Hearer: Head-17. Topic: o25.
 ((visibility v-109)(area v-108)(fill-ratio v-81)) (k i)
   => (k i) (fill-ratio v-125)(intensity v-134)(area v-132))
 [success]
```

A snapshot of the lexicon of one agent is as follows:

meaning	*form*
((visibility v-88))	(d u)
((fill-ratio v-82))	(t e)
((fill-ratio v-81)(area v-86)(visibility v-87))	(l e)
((intensity v-90))	(n a)
((fill-ratio v-81)(area v-86)(intensity v-89))	(p u)
((intensity v-89))	(m i)
((fill-ratio v-81)(area v-108) (visibility v-109))	(k i)

Figure 5 shows the increased success in communication as the agents continue to build up a shared lexicon and the increase in complexity of the lexicons.

Although the physical embodiment of the Talking Heads experiment is quite different from the mobile robots, we see the same phenomena: steady increase and adaptation of a perceptually grounded ontology, and progressive build up and self-organised coherence of a shared lexicon. The Talking Heads experiment is somewhat easier because visual perception provides a richer source for building an ontology and the communication and perceptual conditions are more stable.

6 Conclusions

We consider the experiments so far successful, in the sense that a set of visual categories emerges that are adequate for the language games that the agents play. The categories keep expanding when new segments enter in the environment.

Many extensions and variants need to be investigated further. For the moment only three visual modules and six sensory channels per module are used, this is mainly due to the limitations the black and white cameras pose. Color cameras would produce much more data, which could be used to have more visual modules and sensory channels; providing the lexicon and language formation with extended and richer data.

Second, the visual perception does not receive feedback from the lexicon or language formation. In future work, the higher cognitive processing should return information concerning usefulness of visual modules and sensory channels. This 'selectionistic pressure' could then be used to adjust the way the visual processing is done; e.g. much used visual modules could gain importance, while less used modules would die away.

Finally, rather than a rigid binary discrimination tree, it is conceivable to use a prototype approach in which the outcome of categorisation is based on comparison with a "prototype" point falling inside the possible range of values of a sensory channel. It is still possible to have a hierarchy of more or less general prototypes. Another variant would be to change the task, for example rather than finding a distinctive feature set which identifies what is different between the segments, one could focus on finding what is common between the segments. Such classification games will lead to other visual categories. It is also possible to couple the categorisation component to other user components, for example, a module that does behavioral control based on symbolic descriptions of the scene rather than continuously coupled dynamics. These many variants form the subject of current research and experimentation.

7 Acknowledgements

The design and software simulations of the discrimination games for category formation and the language games for lexicon formation have been developed at the Sony Computer Science Laboratory in Paris by Luc Steels. Several robot builders at the VUB AI Lab (soft and hardware) have contributed to the grounding experiments, including Andreas Birk, Tony Belpaeme, Peter Stuer and Dany Vereertbrugghen. The vision algorithms and the robotic heads were designed and implemented by Tony Belpaeme. Joris Van Looveren has ported the language and discrimination games to the talking heads experimental setup. Tony Belpaeme is a Research Assistant of the Fund for Scientific Research - Flanders (Belgium) (F.W.O.)

References

1. Ballard, D. (1991) Animate Vision. *Artificial Intelligence*, **48** (1991) 57-86.

2. Blake, A. and Yuille, A., editors (1992) *Active Vision*, The MIT Press, MA.
3. Edelman, G.M. (1987) *Neural Darwinism: The Theory of Neuronal Group Selection.* New York: Basic Books.
4. Kohonen, T. (1988) *Self-Organization and Associative Memory.* Springer Series in Information Sciences. Vol 8. Springer Verlag, Berlin.
5. Marr, D. (1982) *Vision* A Computational Investigation into the Human Representation and Processing of Visual Information. W.H. Freeman and Co, San Francisco.
6. Steels, L. (1994) The Artificial Life Roots of Artificial Intelligence. *Artificial Life Journal* 1(1), pp. 89-125.
7. Steels, L. (1996a) Emergent Adaptive Lexicons. In: Maes, P. (ed.) (1996) *From Animals to Animats 4: Proceedings of the Fourth International Conference On Simulation of Adaptive Behavior*, The MIT Press, Cambridge Ma.
8. Steels, L. (1996b) Perceptually grounded meaning creation. In: Tokoro, M. (ed.) (1996b) *Proceedings of the International Conference on Multi-Agent Systems.* The MIT Press, Cambridge Ma.
9. Steels, L. (1996c) Self-organising vocabularies. In Langton, C. (ed.) *Proceedings of Artificial Life V.* Nara, 1996. The MIT Press, Cambridge Ma.
10. Steels, L. (1997a) Constructing and Sharing Perceptual Distinctions. In van Someren, M. and G. Widmer (eds.) *Proceedings of the European Conference on Machine Learning*, Springer-Verlag, Berlin, 1997.
11. Steels, L. (1997b) The origins of syntax in visually grounded robotic agents. In: Pollack, M. (ed.) *Proceedings of the 10th IJCAI, Nagoya* AAAI Press, Menlo-Park Ca. 1997. p. 1632-1641.
12. Steels, L. (1997c) The synthetic modeling of language origins. *Evolution of Communication*, 1(1):1-35. Walter Benjamins, Amsterdam.
13. Steels, L. and P. Vogt (1997) Grounding language games in robotic agents. In Harvey, I. et.al. (eds.) *Proceedings of ECAL 97*, Brighton UK, July 1997. The MIT Press, Cambridge Ma., 1997.
14. Steels, L. and R. Brooks (eds.) (1995) Building Situated Embodied Agents. The Alife route to AI. Lawrence Erlbaum Ass. New Haven.
15. Ullman, S. (1987) Visual Routines. In Fischler, M. and Firschein, O. (eds.) *Readings in Computer Vision*, Morgan Kaufmann Publ., Ca. 1987. p. 298-328.