

Neuromorphic Active Vision Used in Simple Navigation Behavior for a Robot

R. Mudra, R. Hahnloser and R. J. Douglas
Institute of Neuroinformatics, University/ETH Zürich
[regina|rich|rjd]@ini.phys.ethz.ch

Abstract

We explore a neuromorphic robot vision controller which actively extracts sparse information to control the basic navigation behavior of a robot. We investigate the interaction of active vision and the robot's behavior with the objective of avoiding obstacles. A neural map recurrently connected to an active attention pointer [3] uses edge-enhanced visual data as input. The output of the neural network is a population vector representation which contains information about the location of potential obstacles and is used to generate motor control signals. The neural network architecture which we simulated numerically is suitable for an implementation in analog VLSI technology.

1 Introduction

In the last 20 years anatomical, physiological and behavioral studies have suggested three different views of what happens where in the visual cortex [1].

The first view, the anatomical one, is that of ordering visual areas in a hierarchy. The bottom of the (inverted) processing pyramid is the primary visual cortex (V1) and the higher order processes are then arranged in the broad end of the pyramid. These upper areas are involved in progressively more specific processing of the various attributes such as color, form and motion [12].

The second view, the physiological one, separates the visual processing into two streams. One stream is from the M and the other from the P ganglion cells of the retina, which project to the magnocellular and parvocellular layers of the LGN. The P pathway is thought to carry high acuity information about color, texture and form at high contrast, whereas the M pathway is thought to carry low acuity information about low contrast, motion and depth [6], [7], [13].

The third view, the behavioral one, differentiates between two different tasks of vision: In 1969 Schneider postulated an anatomical separation between the visual coding

of the location of a stimulus and the identification of that stimulus [10]. He claimed that two separated cortical visual systems exist. The *What*-pathway specifies the processing information for object identification and corresponds to the dorsal stream. The *Where*-pathway specifies the processing information for spatial localization and corresponds to the ventral stream [11]. Subsequently, a separation of sensory and motor system followed.

The separation into two visual pathways — *What* and *Where* — without connection to the motor system goes against the strong argument that studies of sensory and motor systems should not be separated, which brings us to the question “Where does vision end and action begin?”. It is possible that the rigid separation of brain areas into sensory and motor areas is not useful for understanding the interaction of vision and motor control. Instead, it could be that a complex network of visuomotor modules interconnected with other brain areas is concerned with sensorimotor control [2].

All three of these views demonstrate that as long as we cannot solve the problem of interaction with the world, the answer to the question of interconnection and communication between different brain areas remains open.

In trying to find an answer to the question of how objects must be represented and described we need to think about the use of the representation in interacting with the environment. We wish to explore the connections between sensory and motor systems. A suitable substrate for the investigation of sensory-motor mappings is a robot operating in real-time in a simple constructed environment. A quality measure can be easily formulated from which it is possible to assess the functioning and usefulness of the action oriented perception.

2 Setup with a robot

In the experiment we use the mobile robot *Khepera* designed by K-team (EPFL) [8], [9]. The robot is connected to a host computer via a serial port and the control program

is running under Matlab. The robot has a standard CCD-camera on board tilted downwards at 15° and with a field of view of 60° .

We constructed a *simple* environment of $1.2\text{m} \times 0.8\text{m}$, which contains red, yellow and blue painted wooden obstacles of different shapes in various sizes. The walls and the floor of the environment are white.

3 A basic navigation behavior using contrast information

Underlying the design of the simple behavior arising from edge detection lies the following observation, based on the simple geometrical relationship between distance and the vertical position of objects in the 2D CCD-images. If an object appears at the top or bottom of the image, the object is far away or nearby respectively. The potentially colliding objects lie in an imaginary triangle delimited by the bottom edge and the mid-point of the image. All objects, the lower part (the ‘feet’) of which fall inside this triangle, are potential obstacles for the robot. The objects outside this triangle may only become relevant in the future, when the robot moves forward.

For the *basic behavior* described in this paper, objects are characterized as groups of edges by filtering, compressing and threshold-normalization. First we filter the 210×210 image along both axes, each with a one-dimensional differential filter with the kernel $1/10[-1 -4 -5 0 5 4 1]$. This 1D filtering operation is equivalent to a convolution of the 2D image with a 5×5 Gaussian smoothing kernel multiplied by a 3×3 Prewitt kernel [5]. Both filtered image versions (along X and Y) are then summed. We keep only the absolute value in order to detect positive and negative contrast edges equally. In the next step we compress the image data down to 25×25 pixels by summing over 8×8 sub-matrices. This data is then thresholded at one standard deviation above the mean of the pixel values. The pixels above this threshold are set to one — this operation has the purpose of reducing very high contrast edges in order to prevent these from dominating. Image compression is used to allow the dimensions of the recurrent neural network to be kept small and hence reduce computational cost. This pre-processed binarized image is used as input to a 25×25 map of neurons recurrently connected to an attentional *pointer* [3]. A 1D model of such a pointer-map network is discussed in relation to general aspects of cortical processing such as the relationship between localized and distributed encodings and the attentional enhancement or attenuation of the outputs of clusters of feature encoding neurons [4]. These types of networks have been successfully applied to stereovision and coordinate transformation problems [3].

The input to the attentional pointer biases the processing of the network to the most interesting region in the image.

For the case described in this paper, the network attends to the lowest part of the input to the map, i.e. the ‘feet’ of the nearest object.

3.1 The sensory recurrent pointer-map

The recurrent pointer-map is a neural attractor network which consists of a 2D map M_{xy} of $N \times N$ neurons (where $N = 25$ and $x, y = 1, \dots, N$), two horizontal pointer neurons P_l (left) and P_r (right) and two vertical pointer-neurons P_b (back) and P_f (front). The preprocessed visual input to the map is denoted by m_{xy} and the attentional inputs to the pointer neurons are denoted by p_b, p_l, p_r and p_f . The dynamic network equations are as follows:

$$\begin{aligned} \dot{M}_{xy} = & -M_{xy} + m_{xy} \\ & + \alpha_h(P_l^+ \cos \delta_x + P_r^+ \sin \delta_x) \\ & + \alpha_v(P_b^+ \cos \delta_y + P_f^+ \sin \delta_y) \\ & - \beta \sum_{xy} M_{xy}^+ \end{aligned} \quad (1)$$

$$\dot{P}_* = -P_* + p_* + \sum_{x,y} M_{xy}^+ f(*) \quad (2)$$

The $*$ stands for the four pointer neurons ($* = l, r, b, f$) and $f(*)$ has to be replaced by $(\alpha_h \cos \delta_x)$, $(\alpha_h \sin \delta_x)$, $(\alpha_v \cos \delta_y)$ and $(\alpha_v \sin \delta_y)$ respectively. Neurons have an input-output relationship containing a threshold nonlinearity: $X^+ = \max(X, 0)$. If for a given strength of recurrent inhibition set by the parameter β the strengths of horizontal and vertical excitatory pointer-map connections set by α_h and α_v are not too big, then by existence of a Lyapunov-function every network trajectory converges to a steady state [3]. The strength of vertical coupling is three times stronger than the horizontal coupling and results in the stronger weighting of horizontal edges compared to vertical edges. $\delta_x = \frac{(x-1)\pi}{2N}$ ($\delta_y = \frac{(y-1)\pi}{2N}$) are angles that generate a regular spacing of synaptic weights between map neurons and the horizontal (vertical) pointer neurons.

Figure 1 shows a simplified picture of the network architecture. The 2D map of neurons makes excitatory sine and cosine synaptic weightings (curves above and on the right-hand side of the map) with the four pointer neurons, shown as gray circles. The combination of recurrent pointer-map excitation and global inhibition gives rise to a local soft Winner-Take-All (WTA) mechanism. Neurons on the map close to each other are effectively exciting each other, whereas neurons further away are mutually competing via the inhibition. The inhibitory input $p_b = -5$ to the back pointer ($p_f = 0$) is used as an attentional signal suppressing edges in the back of the image. In Figure 1 the top part of the obstacle is suppressed — it appears in lighter gray than the bottom. This attentional biasing of the network

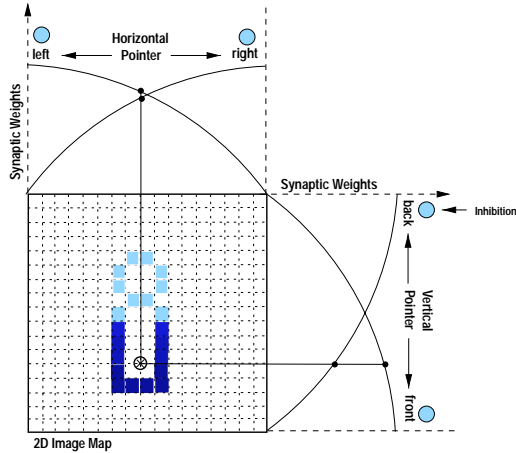


Figure 1. Neural network architecture.

leads to a preference for the front or nearer part of the input independent of the object position in the map (see also Figures 2, 3 and 4). Note that the same effect cannot be obtained by providing a positive (excitatory) input to the front pointer P_f , because given this excitation there would be the emergence of spurious objects in the map in the *absence* of visual input ('ghost' objects). Inhibitory input p_l and p_r to the left and right pointer neurons can be used similarly to bias attention and thus enhance the map output to the right and left respectively. We make use of this option by directing attention to the left during leftward turns of the robot and to the right during rightward turns. This improves and speeds up the reaction to newly appearing obstacles in the visual field.

The four pointer neuron activities P_*^+ are used to read out the coordinates (C_x, C_y) of the closest high contrast edge, the winner of the competition between edges (shown in Figure 1 by the circle on the map):

$$C_x = 1 + \frac{2(N-1)}{\pi} \arctan \frac{(P_r - p_r)^+}{(P_l - p_l)^+} \quad (3)$$

$$C_y = 1 + \frac{2(N-1)}{\pi} \arctan \frac{P_f^+}{(P_b - p_b)^+} \quad (4)$$

The coordinates (C_x, C_y) correspond to the center of activity on the map in steady state given by $\dot{M}_{xy} = 0$ and $\dot{P}_* = 0$.

From a sequence of images (Figures 2, 3 and 4) during which the robot moved in discrete steps towards a group of obstacles, we analyzed the coordinates $(C_x$ and $C_y)$ in order to find rules for navigation. The neural network was reset at each step in order to receive stationary input for simplicity. In each of the figures there are four subfigures. The top-left subfigure corresponds to the original image from the CCD-camera. The top-right subfigure contains the preprocessed

image, which is used as input to the neural network. The bottom-left subfigure shows the output of the map of neurons and the point with coordinates (C_x, C_y) is marked by a gray circle. In the bottom-right subfigure, the dynamics of the 25 neurons belonging to the 12th vertical line of the map are shown (x-axis: timesteps, y-axis: neuron activity, arbitrary units).

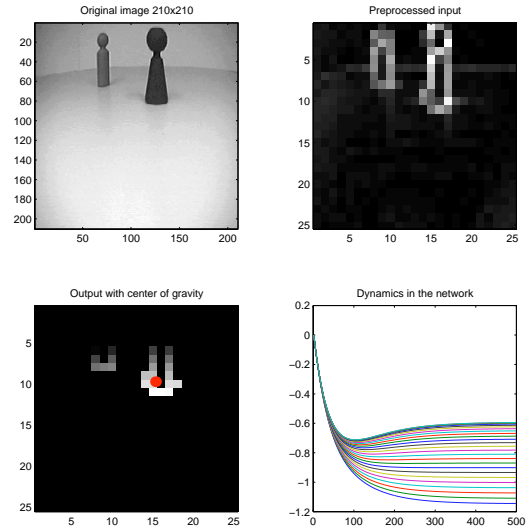


Figure 2. Navigationstep 1.

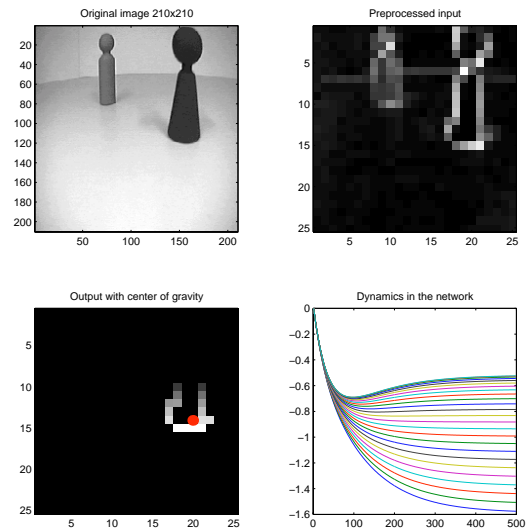


Figure 3. Navigationstep 2.

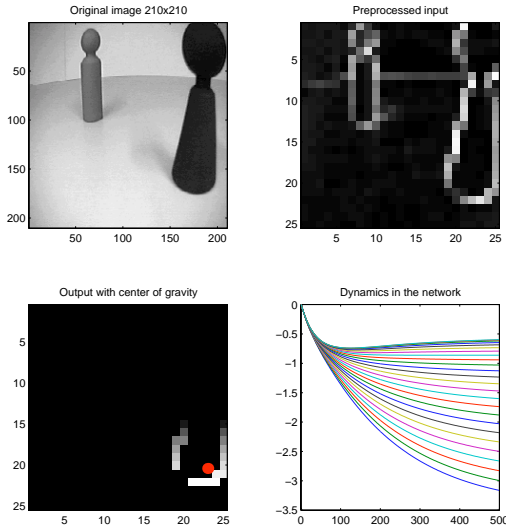


Figure 4. Navigation step 3.

3.2 The connection to the motor-map

To implement an obstacle avoiding behavior for the robot we make use of the empirical triangle determining potential collision obstacles. If the output of the neural network (C_x, C_y) lies in the upper ten horizontal lines of the map, then the speed of the right and left motors are set equally, $w_r = 2, w_l = 2$ (arbitrary units, see Figure 5). If it lies in the lower lines, then the speed of the motors is set to a default value of one. If additionally the point (C_x, C_y) is in the right hand side of the triangle, then the speed of the left motor is set to zero, and conversely if it falls in the left hand side of the triangle, then the speed of the right motor is set to zero.

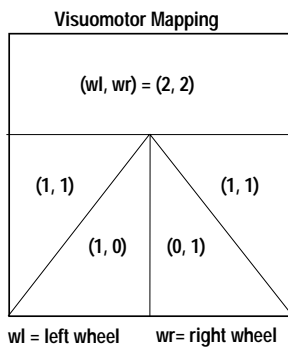


Figure 5. Motor-map.

With these motor settings the robot moves straight forwards as long as no obstacle appears in the triangle. However, when an obstacle is present then the motor signals de-

crease. By selectively decreasing or zeroing the speed of the left and right motors, the robot is able to successfully avoid obstacles.

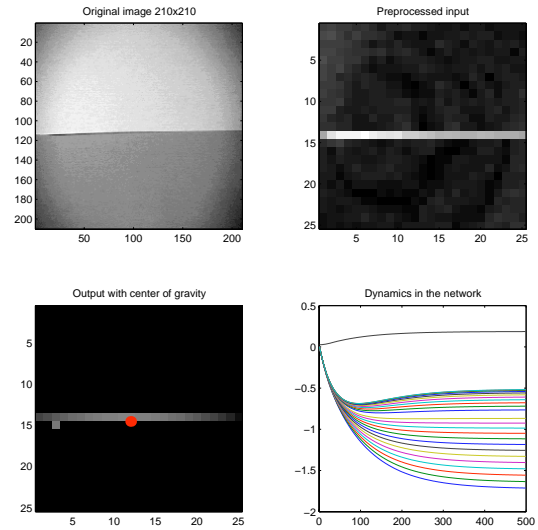


Figure 6. Empty environment.

If there are no obstacles (Figure 6), the only line in the image is the edge between the wall and the floor of the environment. The focusing of attention on this edge prevents the robot from running into the wall of the environment, according to the same avoidance behavior as in the presence of obstacles.

Figure 7 shows the environment with the robot and a sketch of the paths taken by the continuously moving robot avoiding obstacles. Most of the time the robot does not run into the obstacles. However, there are situations in which when the robot, already close to a wall, tries to avoid an obstacle, it bumps into the wall. The explanation is that the robot has a dead zone in its immediate front. So if it turns towards the nearby wall away from an adjacent obstacle, the edge between wall and floor is already outside its visual field. To avoid such situations the robot would have to have short-term memory or richer sensory input.

Note that it would not make sense to compute the population vector directly on the visual input image and connect it to the motor-map, because only the lower parts of objects are relevant. The attentional neural network computes the necessary nonlinear focusing of map activity onto the lower part of the input.

4 Conclusion

We have explored the interconnection of a visuomotor system by implementing a particular neuromorphic representation of objects in the environment, and we have gen-

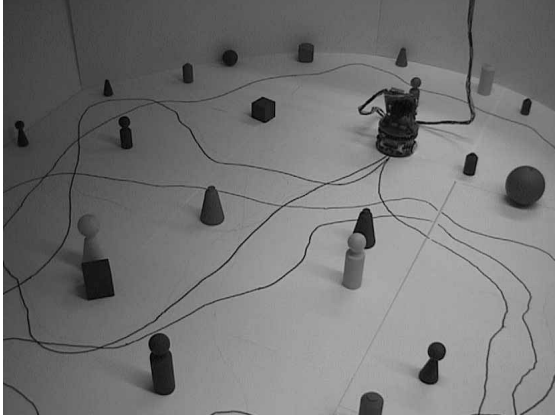


Figure 7. Paths of the robot in the environment.

erated a reasonably competent, obstacle avoiding, simple navigation behavior. We have shown with our experiment that a recurrent neural pointer-map can allow a robot to move around in a simple environment by producing an active population vector representation of relevant visual information. The architecture of linear threshold neurons is suitable for aVLSI implementation. The summing operation of the neuron's output current can be implemented by Kirkhoff's current summing law and the rectification non-linearity and synaptic interaction of neurons can be realized by current mirrors [3].

We plan to improve the overall navigation skills of the robot by designing additional basic behaviors which use further sensory modalities together with additional instances of the same types of recurrent neural networks with attention pointers.

Acknowledgments

This work was supported by the Swiss National Science Foundation SPP Grant. We also thank A. W. Whatley and G. Indiveri for the correction of the manuscript.

References

- [1] R. J. Douglas, K. A. C. Martin, and J. C. Nelson. The neurobiology of primate vision. *Bailliere's Clinical Neurology*, 2:193–225, 1993.
- [2] M. A. Goodale. Visuomotor control: Where does vision end and action begin? *Current Biology*, 8:489–491, 1998.
- [3] R. Hahnloser. *Computation in Recurrent Networks of Linear Threshold Neurons: Theory, Simulation and Hardware Implementation*. PhD thesis, ETH, 1998.
- [4] R. Hahnloser, R. J. Douglas, and K. Hepp, 1998. Submitted to *Nature Neuroscience*.
- [5] M. Lemmens. *Structure-Based Edge Detection*. PhD thesis, ETH, 1996.

- [6] D. H. Livingstone, M. S.; Hubel. Psychophysical evidence for separate channels for the perception of form, color, movement, and depth. *Journal of Neuroscience*, 7:3416–3468, 1987.
- [7] D. H. Livingstone, M. S.; Hubel. Segregation of form, color, movement, and depth: anatomy, physiology, and perception. *Science*, 240:740–749, 1988.
- [8] K.-T. S.A. Khepera: General I/O turret user manual, v 2.0. Technical report, EPFL, Lausanne, 1995.
- [9] K.-T. S.A. Khepera: User manual, v 4.06. Technical report, EPFL, Lausanne, 1995.
- [10] G. E. Schneider. Two visual systems. *Science*, 163:895–902, 1969.
- [11] L. G. Ungerleider and M. Mishkin. Two cortical visual systems. In D. J. Ingle, M. A. Goodale, and R. J. W. Mansfield, editors, *Analysis of Visual Behavior*, chapter 18, pages 549–586. MIT Press, 1982.
- [12] J. H. R. Van Essen, D. C.; Maunsell. Hierarchical organization and functional streams in the visual cortex. *Trends in Neuroscience*, 6:370–375, 1983.
- [13] S. Zeki, S. M.; Shipp. The functional logic of cortical connections. *Nature*, 335:311–317, 1988.