Presented at EXPO 2000: Shaping the Future, Hanover, Germany, July 11-13th 2000

A Multi Agent Approach to Machine Vision

Stavroulakis S, Callaghan V, Spacek L
University of Essex

Abstract. *Biologically inspired behaviour-based approaches to agent design have been particularly successful in mobile robotics. In this approach, rather than decomposing intelligent systems by function, the system is segmented into independent components of end-to-end functionality, which mirror behaviours such as wall following and obstacle avoidance. To date, although vision has been a vital part of robotic systems, it has tended to be treated as a monolithic peripheral, even in behaviour-based agents. This research explores architectures for systems that extend decomposition by behaviour to vision. In such platforms a number of camera agents, each monitoring some simple vision characteristic (e.g. movement or boundaries) learn to collaborate in real time on the interpretation of a scene. Visual learning unifies methods that are developed under the major dichotomy between supervision and adaptation in machine learning. Two monochrome non-stereoscopic look-forward cameras are used to test the strength of agent-based vision for mobile robots.*

1 Introduction

Biologically inspired integration of vision functionality has been suggested in different domains [15,2], but little has been done on integrating vision methods for situated agents [19] as potential carriers from specialised applications to multidomain intelligence [1,5,7]. It is also noticed [9] that a comprehensive theory that collects isolated algorithms for image understanding into a functional vision system using space state representations is missing.

Other architectures use vision behaviours to control active heads [3], or as both tools for vision integration and controllers for motion behaviours [4]. Integration of vision functionality is possible through behaviours, drawn from biological vision and behaviour-based robotics in an attempt to bridge the gap between AI and specialisations of computer vision [12].

Performance improvement is built around a base group of co-working vision methods that the robot learns to operate in real time. Both supervised learning and adaptation are used, since there is no dominant neural structure to enforce either one or the other. The model deals with the hierarchy of competences that subsumption does not provide [11].

2 Vision behaviours

2.1 Motion detection

The motion segmentation module quickly locates activity on image parts by independently estimating the direction and location of the motion field on a region array representation of successive images For fast estimations the motion field is expressed by a set of velocity vectors, each one assigned to a region of neighbouring points.

In taking decisions about where to turn according to the direction of a moving object, the 2D motion field is sufficient, rather than producing a complete optic flow. Vector values corresponding to motion, no motion and insisting motion are assigned to regions. The normal flow is used to decide a robot' s behaviour [16,20]. In [20] it is asserted that if the moving object is within the visible field and its direction takes two subsequent images to be decided, the preference for the complete optic flow over the simple computation of the location of the motion vectors does not give any significant advantage to what action the robot has to take.

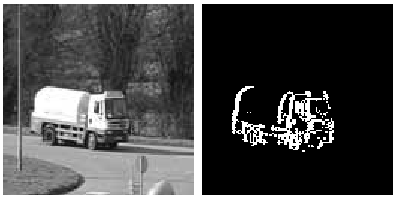


Fig. 1. Simple subtraction of point values for motion segmentation

2.2 Edge detection

Edge-based segmentation exploits the knowledge offered by motion-based methods, to retrieve the location of object bounds. A Sobel operator of *5x5* horizontal and vertical masks is applied locally to every image region. Pixels around the centre are double weighted to highlight significant discrepancies. By restricting the edge operator to small regions of the image, decisions on steering behaviours can be taken even before the process is completed. The masks apply to an image region and the region is scanned either horizontally or vertically, depending on which mask outnumbers the results of the other, to examine the magnitude differences $magn_{ij} = \sqrt{f_x^2 + f_y^2}$ of neighbouring points against an automatic threshold T. If B is an image and

$$\begin{aligned} a_1 &= |magn_{ij} - magn_{ij-1}| \\ a_2 &= |magn_{ij} - magn_{ij-2}| \\ a_3 &= |magn_{ij-1} - magn_{ij-2}| \end{aligned}$$

Table 1. The Sobel masks

-1	-1	0	1	1	1	1	2	1	1
-1	-1	0	1	1	1	1	2	1	1
-2	-2	0	2	2	0	0	0	0	0
-1	-1	0	1	1	-1	-1	-2	-1	-1
-1	-1	0	1	1	-1	-1	-2	-1	-1
		f_x					f_y		

then segmentation values are defined by the condition:

$$B_{ij} = \begin{cases} 0 & (a_1 \leq a_2 \text{ and } a_2 \geq T) \text{ or } (a_1 = a_3 \text{ and } a_3 \geq T) \\ 1 & (a_1 > a_2 \text{ and } a_1 \geq T) \end{cases} \tag{1}$$

After the image scanning, regional peak values are compared with a second threshold to subtract noise from edges. A high threshold reduces noise, however results in missing edges. The automatic threshold, in a similar approach to [17], is unique for every region and is decided through a selection of values from a vector of magnitude differences. First, the value with the highest frequency of occurrence is selected as a low bound and then the set of all values bigger than the low bound is searched. The value with the lowest frequency is selected. The result is the second or third best threshold value near the optimum as it has been found in simulations. The threshold value retreats from optimum as the number of edges increases. Regions with edges appear to be brighter. For those regions, foreground and background intensities are clearly separated from each other. The second threshold holds for all regions within the agent's visual field and its values are experimentally decided for different regions' resolutions.

2.3 Statistical indices

The image is divided into four rectangular regions. Average intensities are calculated for each region, as well as for the whole image. An estimation of the average intensity of the lower left and right regions, as well as the upper left and right regions of the image, is a first hint of either the presence of an object or of free space. The robot, guided by regional dissimilarities in light intensity, adjusts its steering behaviour. However, a change in steering, to avoid the uncertainty introduced by the light variations, does not take place instantaneously. The delayed reaction depends on the relative persistence of the light pattern. Those indices are the simplest to use in appearance-based methods for a first classification of surface features.

3 The learning component

As suggested in [6], we explore habituation and sensitisation as types of learning. For Thorpe, habituation is one of the categories of animal learning [14].

In [13], habituation, and in [8], habituation and sensitisation become part of a neural model of specific functionality. For our study, habituation is a situated discriminatory behaviour of vision-based learning, quantified by two parameters:

- the reduction of the standard deviation $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \tilde{\mu})^2}{n}}$ of the robot's average trajectory distance measurements $\tilde{\mu}$ from a lane or wall in object following, which certifies the stability of the specified navigation task
- the reduction of the relative standard deviation $RSD = 100 * \frac{\sigma}{\tilde{\mu}}$, which is an index for the reduction of supervisory signals and ensures assimilation of behaviours.

The learning component is a vision method with the additional capability of updating the base group of vision methods with respect to their output types and it provides an ideal example of how sensitisation can be defined for behaviour-based systems. It is also similar to the method of average intensities. Both methods have the same output types and divide the image into 4 parts to compute partial average intensities. The difference between them is the threshold value and the difference in the steering angle. It turns out that despite their similarities they are different methods, because the behaviours they suggest are not identical. The learning method is designed either to avoid dark areas or to report night conditions in case no region's intensity is above the night threshold. In case it reports dark conditions for one of the agents, it causes the base group to be updated according to a fixed update strategy that adds a single method to the group. In case the tutor partially covers the cameras' lenses, the method only steers the robot in the direction that corresponds to bright regions, without applying learning.

A pool of methods is the source for feeding the base group with a variety of combinations. In case learning neither adds nor replaces methods in the base group, the agent has stabilised its behaviour in a particular environment. In case the environment is altered and steering is not as expected, the tutor causes the learning method to continuously update the base group until the robot's behaviour is re-stabilised.

Being part of the base group, each vision method's output value suggests a particular behaviour to be adopted by the robot's wheels. A decision mechanism, based on priorities of output types, serialises suggestions of vision methods. If random priorities are used, a voting classifier for selection of subgroups is necessary. Our priorities are ordered, a choice that promotes cooperation over competition. Failures to generalise are corrected by the tutor. Decisions on which camera will execute which behaviour are taken by one of the agents, which has the role of manager and collects suggestions from other agents through message exchange. Alternatively, the model has been tested with a separate manager process, allowing to the camera agents indirect communication only. Learning has absolute priority over steering behaviours. Steering codes are preferred to behaviour codes for wall following, obstacle avoidance and tracking in order to preserve behaviour slicing in smaller parts common to all behaviours. Thus, behaviours defined by low-level assembled components are used for scaling to determine behaviours that were not part of the original design. The selected behaviour is the one with the highest priority in the base group. Each method also advocates the way the suggested behaviour will be implemented. As a consequence of having different output types, each method may propose a different implementation of the same behaviour. In that case, a decision made by minor rules is taken on which method will activate the suggested behaviour. Introduction to the pool of a newly designed vision method is possible in the following five steps, by loading a relevant object module from a host machine:

- declaration of the function prototype
- creation of a new pointer in the list of methods
- initialisation of the function's output type
- writing the code for function activation
- writing the behaviour rule

A design decision for supervised learning to be applied to one method to affect the disposition of the base group rather than being applied to each individual method or to a manager agent for a particular hierarchy of agents is related to the role of agent managers. In our architecture, managers are assembly points for signals that evaluate an action. Managers do not send signals back to individual methods. This sort of feedback is allowed for the learning method only. Individual methods have their own means of adaptability by adjusting their parameters according to brightness patterns. Supervision, rather than being applied directly, is spread to base group members through the learning method, which is a locus of control, a 'gate', unique for each embedded agent, which decides for the behaviour of the robot or the sensory agent. After the 'gate' is stimulated, re-assigning of methods in the base group is decided according to the implemented strategy. Thus, control is limited to a single stimulus or gesture.

4 Experiments

Two indoor environments are used separately: a row of blue cylinders of $50cm$ height and an L-shaped wooden wall $30cm$ high of virtually no contrast with the floor. The x_i distance measurements are taken from the point where the perpendicular to the convex hull meets the actual trajectory trace. The robot goes around the blue cylinders first, using only the normal flow for two rounds before a learning signal adds a vision method based on average brightness, which reduces σ values significantly. With this active configuration of vision methods, no further supervision is required. It is not necessary to set the robot at an accurate distance from the wall, since convergence to a trajectory at a $50cm$ average distance away from wall is expected.

After the supervisor visually guides the robot to the L-shaped wall, the robot goes around the wall, using the current group of vision methods. After completion of the first round, an edge detector is added. The improvement in this case is not drastic because the edge detector converges to the path less smoothly to ensure that in case of corners the robot is not receding too far away from the wall.

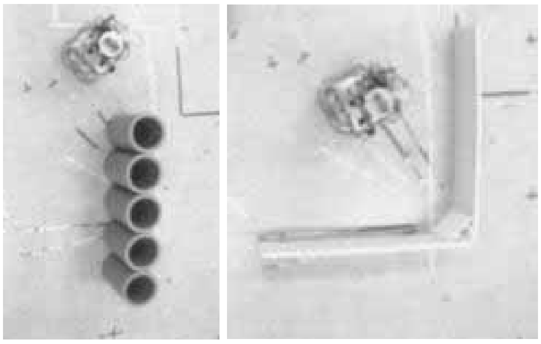


Fig. 2. A view of the indoor environments

The resolution of the image is $128x128$. A $16x16$-array mask of 8x8 regions is a compact form of a higher-level view of the original image and it is used as a measure of the surface depth from the robot. Simulation shows that this is the optimal division of regions for a motion detector and for the given image size. A bigger size does not allow noise to be easily separated from real edges. Smaller size almost reproduces the original image into a low abstraction level not appropriate for calculations. Obstacle avoidance emerges as part of a line following behaviour. Navigation is stable, although the errors from the angle measurements can be frequent and sometimes insistent. The turning round object behaviour is affected by shadows and an on board light source may solve the problem, at least for indoor environments, in case the sharpness of shadows is constant. A delay regulates the time-to-turn so the robot does not turn too close to the vertical edge of the wall. Acceleration of turning is also applied to ensure the robot will re-establish contact with the wall. Errors in intensity differences due to illumination and low pixel resolution are handled effectively by dynamic thresholds, so the robot keeps a minimum and a maximum distance while going around the blocks. The position and angle of the fixed on-board camera implicitly decide the minimum distance. The turning behaviour and the presence of obstacles decide maximum distance.

Table 2. The codes for the vision methods

Vision method	*code*
learning component	1
normal flow	2
statistical indexes	3
Sobel edge detector	4

Table 3. Distance measurements of the robot's trajectory trace from a wall of blue cylinders with normal flow and the learning component as members of the base group of vision methods.

	average $\tilde{\mu}$	*variance* σ^2	*standard deviation* σ	*relative deviation* *RSD* %
first round left side	51.75	297	17.2337	33.3018
first round right side	41.75	100.75	10.0374	24.0417
second round left side	47.6875	89.1836	9.4437	19.8033
second round right side	42.5	38.25	6.1847	14.5522
complete first round	46.75	223.875	14.9625	32.0053
complete second round	45.0938	70.4443	8.3931	18.6125

Table 4. Trajectory distance measurements for the blue cylinders environment after average brightness is added to the base group of vision methods.

	average $\tilde{\mu}$	*variance* σ^2	*standard deviation* σ	*relative deviation* *RSD* %
first round left side	62.375	30.4844	5.5213	8.8518
first round right side	55.4375	8.9648	2.9941	5.4008
second round left side	55.3125	9.6836	3.1118	5.6258
second round right side	55.3125	18.9336	4.3513	7.8667
complete first round	58.9062	31.7568	5.6353	9.5666
complete second round	55.3125	14.3086	3.7827	6.8388

Table 5. Trajectory distance measurements for the L-shaped wall.

	average $\tilde{\mu}$	*variance* σ^2	*standard deviation* σ	*relative deviation* RSD %
concave side codes 1,2,3	52.33	296.8889	17.2305	32.9266
convex side codes 1,2,3	51.6818	70.1942	8.3782	16.2111
complete round codes 1,2,3	52.1591	186.4406	13.6543	26.1782
concave side codes 1,2,3,4	56.1111	108.3210	10.4077	18.5484
convex side codes 1,2,3,4	44.9167	60.7014	7.7911	17.3457
complete round codes 1,2,3,4	49.3478	114.9877	10.7232	21.7298

5 Discussion

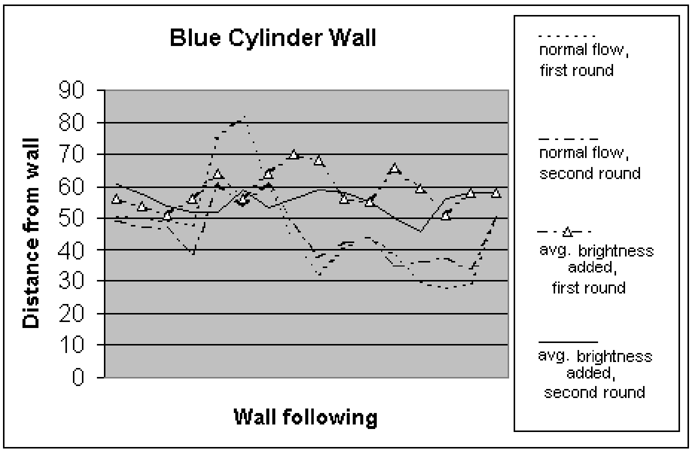
Incomplete vision processes at various levels ought to be used in conjunction with each other. It is an approach that seeks additional evidence for the interpretation of a scene in the teamwork of vision methods. We choose to combine motion and edge detection, which are methods of solid foundation in computer vision, and to refrain from the requirement of 100% accuracy, changing from a methodology of visual cues to direct perception. If the trajectory deviates from a specified angle value a line following function steers the wheels in the opposite direction. Thus, this function is used both for wall following and obstacle avoidance. Vision functions have been used before as multi-purpose processes. [10] uses the Hough transform on congruent triangles and alternatively on congruent polygons to detect planar motion.

We add learning to deal with the complexities and the potential high population of vision processes. Our results show that vision-based learning applies well to incomplete vision processes by smoothing the distance variations of the robot from a wall.

6 Conclusions and future work

Our approach suggests cooperation of simple vision behaviours that run in parallel, each one having its own functionality along the path from input to output. These vision agents, based on noisy, low-resolution data and incomplete error-prone vision methods, learn to govern a robotic platform.

The introduction of new vision methods and a formal description of the learning model will be part of the future work.



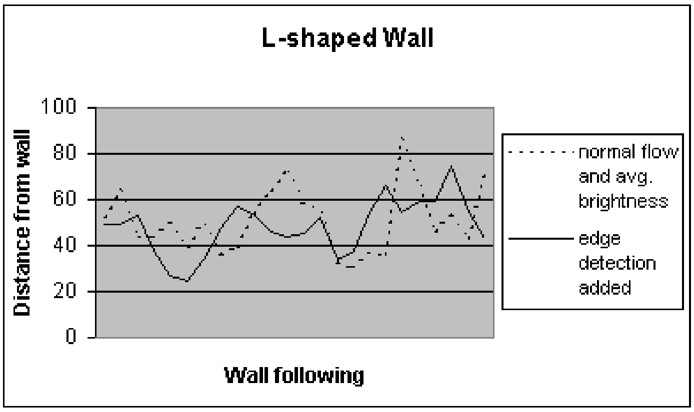


Fig. 3. Robot's trajectory becomes even as the combination of vision methods is updated.

References

1. Y. Aloimonos, Visual Navigation, (1996) Erlbaum, Lawrence Associates, Inc. ISBN: 0805820507
2. Claus S. Andersen, Stephen D. Jones, and James L. Crowley, Appearance based processes for visual navigation. In Henrik I. Christensen, editor, Proceedings of the International Symposium on Intelligent Robotic Systems, Stockholm, Sweden, July 1997 http://citeseer.nj.nec.com/andersen97appearance.html
3. J. Batista, P. Peixoto, H. Ara'ujo, Real-Time Visual Behaviors with a Binocular Active Vision System, ICRA97 - IEEE Int. Conf. on Robotics and Automation, Albuquerque, New Mexico, USA, April 1997 http://citeseer.nj.nec.com/146621.html

4. Alexandre Bernardino, Jos Santos-Victor, Binocular Visual Tracking: Integration of Perception and Control, IEEE Transactions on Robotics and Automation, (15)6, Dec. 1999 http://citeseer.nj.nec.com/301124.html
5. M. Brand, L.Birnbaum, P. Cooper, Sensible Scenes: Visual Understanding of Complex Structures through Causal Analysis, Proceedings, National Conference on Artificial Intelligence, Washington D.C., 1993 http://citeseer.nj.nec.com/brand93sensible.html
6. R. A. Brooks, From Earwigs to Humans, in Robotics and Autonomous Systems, Vol 20, June 1997, pp. 291-304 http://citeseer.nj.nec.com/brooks96from.html
7. Amit Chakraborty, James S. Duncan, Integration Of Boundary Finding And Regionbased Segmentation Using Game Theory, IPMI 1995 http://citeseer.nj.nec.com/chakraborty95integration.html
8. R. I. Damper, R.L.B. French, T.W. Scutt, ARBIB: an Autonomous Robot Based on Inspirations from Biology, in Robotics and Autonomous Systems, 31 p.247-274 http://citeseer.nj.nec.com/damper99arbib.html
9. B. Draper, Learning Grouping Strategies for 2D and 3D Object Recognition, In ARPA Image Understanding Workshop. Palm Springs, CA, 1996 http://vis-www.cs.umass.edu/publications/pubs-1996.html
10. Iris Fermin, Atsushi Imiya, Akira Ichikawa, Two probabilistic algorithms for planar motion detection. IEICE Transactions on Information and Systems, VOLE80D, no 3, pp. 371-381, 1997 http://search.ieice.or.jp/1997/files/e32006.htm
11. C. Fermüller and Y. Aloimonos, Vision and Action, Image and Vision Computing, 13 (10), 1995, 725-744 http://citeseer.nj.nec.com/135622.html
12. Fisher, R., Is Computer Vision Still AI?, AI Magazine, Vol 15, No. 2, pp 21–27, 1994. http://vision.dai.ed.ac.uk/rbf/publications.html
13. S. Marsland, U. Nehmzow and J. Shapiro, A Real-Time Novelty Detector for a Mobile Robot, EUREL European Advanced Robotics Systems Conference, Salford 2000 http://citeseer.nj.nec.com/marsland00realtime.html
14. Donald Michie, Trial and Error (1961), in On Machine Intelligence, 2nd edition, Chichester : E. Horwood ; New York : Halsted Press, 1986
15. R. Milanese, H. Wechsler, S. Gil, J-M. Bost, T. Pun, Integration of Bottom-Up and Top-Down Cues for Visual Attention Using Non-Linear Relaxation, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1994. http://citeseer.nj.nec.com/milanese94integration.html
16. Ulrich Nehmzow, Vision Processing for Robot Learning, J. Industrial Robot, Vol 26, No 2, 1999, ISSN 0143-991X http://citeseer.nj.nec.com/nehmzow99vision.html
17. T. Rachidi, L. Spacek, Constructing coherent boundaries, Technical Report, University of Essex, Computer Science Dept., Dec 1994 http://cswww.essex.ac.uk/technical-reports/1993-94.htm

19. L. Steels, R. Brooks, The artificial life route to artificial intelligence : building embodied, situated agents, Hillsdale, N.J. : L. Erlbaum, 1995
20. Lucia Vaina, The Magic Hat: an adaptive solution for enhancing visually guided mobility of the visually impaired http://www.bu.edu/eng/labs/bravi/Research/MagicHat/magicHat.htm