Inducing the focus of attention by observing patterns in space

Derek Magee and **Chris Needham**
{drm,chrisn}@comp.leeds.ac.uk
School of Computing
The University of Leeds
Leeds, LS2 9JT, UK

Paulo Santos*
psantos@fei.edu.br
IAAA
Centro Universitario da FEI
Sao Paulo, Brazil

Sajit Rao
sajit@dist.unige.it
DIST
University of Genoa
Genoa, Italy

Abstract

This paper builds on existing work on *learning protocol behaviour from observation* to propose a new framework for visual attention. The main contribution of this work resides in the fact that attention is not given a *priori* to the vision system but learned by induction from the active observation of patterns in space. These patterns are sequences of coloured objects that are placed by an agent in a camera field of view. Therefore, in this work we propose a method for learning the focus of attention from the visual observation of tasks executed by an agent. The description of objects in space in terms of observer-object relative frames of references, named *local cardinal systems* is a second contribution of this work.

1 Introduction

The subject of visual attention has been one of the most neglected themes in computer vision and image understanding. As surveyed in [Tsotsos, 2001], authors have been making strong assumptions about attention in order to develop other issues in computer vision. Assumptions such as: one-to-one correspondence between figures in adjacent frames [Siskind, 1995]; regions of interest in the image manually given as inputs [Mann *et al.*, 1997][Bobick, 1997], etc. A few authors have proposed models for predicting where to search for corresponding regions from image to image [Tsotsos, 1985][Shanahan, 2002][Dickmanns, 1992][Baluja and Pomerleau, 1997]. However, the problem of how such expectancy models could be (themselves) automatically learned from the visual observation of tasks has not yet been addressed.

This paper describes some preliminary steps towards the construction of a computer vision system that is capable to automatically induce the focus of attention from the visual observation of patterns being created in space. In the current stage of this work, patterns in space are formed by coloured blocks that are stacked by an agent in such a way to create repetitive arrangements of colours. An active vision system tracks the blocks as they are being stacked. Data from this system feeds an inductive logic programming system (ILP) that generates a *model of expectancy* about which object should be placed and in which position. This provides the basis for a spatial attention mechanism with which an autonomous agent could predict where (and what) to expect in a particular region of space or where and what to place a particular object. Therefore, the resulting model of spatial expectancy is learned from the observation of agents acting in the external world.

A secondary contribution of this paper is the definition of a new framework for representing qualitative spatial relations to be input in the ILP system. A central element in this representation is a observer-object relative cardinal reference frame, named the local cardinal system.

This paper is organised as follows. The next section presents the active vision system that forms a first layer for visual attention. Visual attention is introduced in Section 3; how the focus of attention is induced from visual observation is discussed in Section 4. The local cardinal system is defined in Section 4.1.

2 Active Vision

In this section we briefly explain some of the components of the active vision system which is used in this work to control a pan-tilt camera[1] and to provide data for inductive learning the focus of attention of an active observer. Currently no robotic hands are used for moving objects, such as those used in previous work [Fitzpatrick *et al.*, 2003], although this is an extension we wish to make.

The Active vision process can broadly be divided into four components:

1. **Bottom-up colour and motion saliency:** that are measures applied everywhere in the image.
2. **Figure-ground operations:** incorporating some top-down biases that are applied only at the focus of attention.
3. **Active tracking:** uses figure-ground operations.
4. **Marker tracking:** maintains a short-term memory of interesting locations by using a combination of proprioceptive transformation and figure-ground operations.

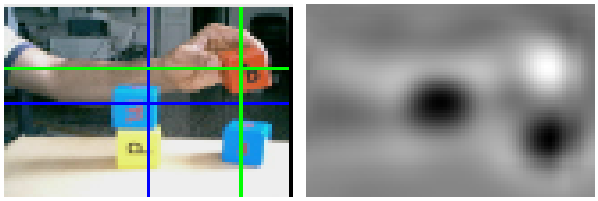
These processes are discussed in turns bellow.

*Corresponding author.

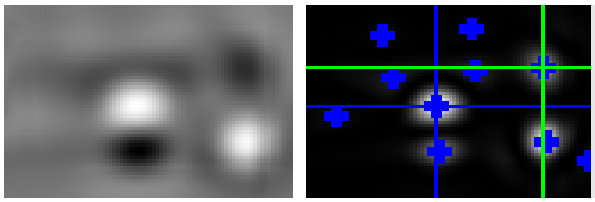
[1] A logitech *QuickCam Orbit*.

2.1 Colour and Motion Saliency

Colour saliency is computed by applying a difference of Gaussians (DOG) filter on the Red-Green (RG) and Blue-Yellow (BY) channels, and taking the sum of the squares of the results. The local-maxima of DOG filters applied to images indicates the centres of interesting blobs. Figure 1 shows an example of the original image, the RG and BY DOG responses, and the sum squared image. The crosses in Figure 1(d) mark local-maxima in the sum-squared image, i.e. centres of salient blobs.



(a) The original image (b) RG DOG



(c) BY DOG (d) Sum Squared

Figure 1: Colour Saliency

Motion saliency is computed by two measures: one is a correlation-based optical flow followed by a connected-components and bounding box computation; and another is the temporal derivative of the colour-saliency measure. The second measure is more selective than the first because it is tuned to *moving* coloured blobs at a particular scale (set by the scale of the DOG filter).

The results of computing colour and motion saliency are used for motion triggering. Motion triggering is the process by which the system decides whether a particular movement is worth saccading to, and tracking or not. Naturally, this is a top-down application dependent criterion. In this case, only the movement of well-defined coloured blocks is worth tracking. The blob centre closest to the centre-of-mass of the colour-salience, temporal derivative is chosen as the most likely-location for the centre of the moving blob.

2.2 Figure-ground at focus of attention

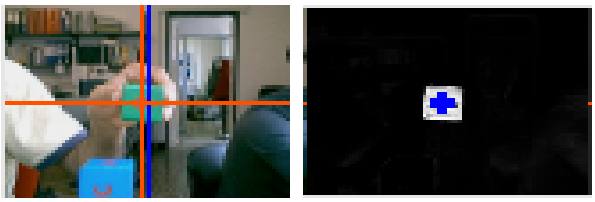
The figure-ground processes are always applied at the centre of the image, as well as at any motion triggering points, if they exist.

The steps in doing figure-ground are: the selection of a colour that is present in the fovea (a small disc around the centre of the image); the application of the DOG filter on the selected colour image; and a spreading-activation in the zero-crossing image from the centre. The result of the spreading activation operation enables the system to decide whether the object is well-bounded or just a background feature (in which case the activation colours the entire image).

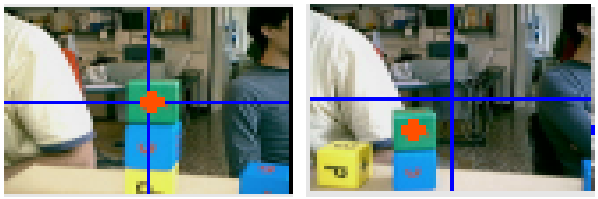
Therefore, the focus of attention does not always need to be at the centre of the image. It could, for instance, be the result of figure-ground operations *at the triggering point*. This helps in the decision of whether there is a well-defined block at the triggering location which is worth tracking.

2.3 Active Tracking

Once an object at the triggering location is found to be a well-defined block, the system sets the tracking model to be the characteristics of the block (colour and size) and immediately saccades to that location. Now, given that the system has a model of what it is supposed to be tracking, it can find the best match to that model near the centre of the image and saccade there.



(a) Active tracking (b) Object colour mask



(c) Marker creation (d) Marker tracking

Figure 2: Active tracking

Figure 2 shows an example where the system triggers on and saccades to the block that suffered motion. Panel 2(a) shows the focus of attention following the object in motion (i.e. the camera actively tracks the object); panel 2(b) shows the creation of a colour mask for the object; the marker creation is depicted in panel 2(c) and in 2(d) a marker is tracked as the camera pan and tilts to focus somewhere else, keeping the previously created mark.

It is worth noting that the tracked object needs only to be salient to trigger the tracking (i.e. attract attention) but once the system starts using the model of the object, the tracking is robust and does not rely on the bottom-up saliency map at all.

2.4 Marker Creation and Tracking

Once the tracked object comes to rest, a marker is dropped on the object. A marker has the function of short-term memory that binds "what" and "where" during a task (in effect, several times during the same task). Marker positions therefore have a retinal component as well as a proprioceptive component. Being important to the task, markers are always tracked, no matter what else the system is doing. If a marked object goes out of the camera's sight, tracking this object's marker stops, resuming as soon the object comes back in view. This is possible only because marker's state includes proprioceptive information as well as a model of the object (the object's signature). Marker tracking is therefore a combination of proprioceptive to retinal transformations followed by active tracking using the model of the marked object.

3 Attention

In this work, three levels of attention are used to schedule visual resources: data driven visual attention (A1), statistically learned spatial attention (A2), and symbolically learned spatial attention (A3). Both A2 and A3 are expectation driven models which are learned from prior information. These levels of attention combine to provide *task driven attention*, since they are learned from prior observation of the typical interactions of a human with a task at hand.

As the main interest of this work is (A3) we briefly overview attention mechanisms (A1) and (A2), and concentrate on the discussion of the symbolically learned spatial attention in Section 4.

3.1 Data driven visual attention (A1)

Data driven visual attention comprises exactly the active vision system described above. In an embodied agent, with a restricted field of view, the camera needs to be able to move to explore the whole of the scene. Therefore, the focus of attention should follow salient objects through the scene.

In this level of attention, the set of markers are associated with salient (interesting) objects to which the focus of attention is directed to.

This primary attention mechanism provides the means for the process of learning spatial attention by the visual observation of tasks, as discussed in Section 4 below.

3.2 Statistically learned spatial attention (A2)

A probability density function in the form of a particle distribution of prototype vectors is learned using an unsupervised competitive learning neural network [Johnson and Hogg, 1996], which is essentially an online vector quantisation method. This is used to move the camera to places where objects are expected to be, and the active vision system is used to locate the object, or to report that there is no object at the location.

This attention mechanism competes with (A3) in order to provide an accurate model for the focus of attention. Research on the interplay between statistically and symbolically learned attention is well under way.

4 Symbolically learned spatial attention (A3)

In this section we discuss how spatial attention can be learned from observation of examples of activity.

This work uses inductive logic programming to learn patterns in the space formed by object positions and colours. For instance, if the system observes a human agent building a tower of coloured blocks defining a repetitive pattern of colours (e.g. blue blocks on top of red blocks and vice versa), the symbolic learning should be able to find a set of rules with which a synthetic agent could predict which would be the next block on the tower, and its position in it. For this end, we need to define a spatial frame of reference and an appropriate set of spatial relations to describe the states of the world.

As frame of reference we define an observer-object relative reference system, named *the local cardinal system* (LCS), which is discussed in Section 4.1. The set of spatial relations assumed in this work is introduced in Section 4.2, while the symbolic learning executed with this representation system is presented in section 4.3.

4.1 Representation: the local cardinal system

The observer-object relative frame of reference used to describe objects in space in this work is shown in Figure 3.

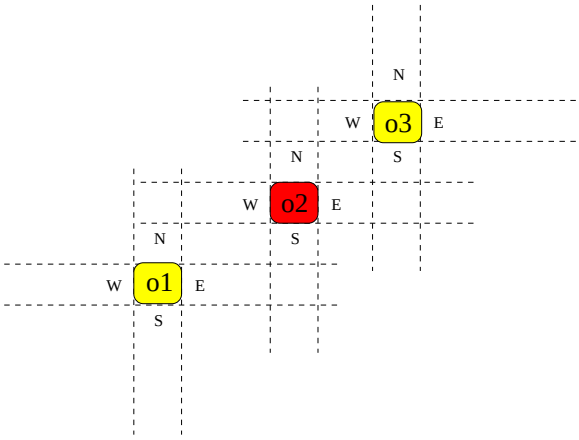


Figure 3: The local cardinal system.

The local cardinal system works in the following way, each object that is placed on the table defines its own cardinal reference frame that is used to describe the other objects around it. An object is only described within the nearest frame of reference available. Moreover, as a simplifying assumption, an object is only described within the reference frame of another if the former is *placed* after the latter. I.e., objects already placed on the table are not described in the reference frame of newly placed ones.

Therefore, the local cardinal system is object relative, as each object is represented with respect to another, but also observer relative as the cardinal directions will be dependent on the observer's viewpoint[2].

[2]For instance, the direction north w.r.t. an object will always point in the same direction as the observer's gaze.

In order to avoid ambiguous descriptions when an object falls on the threshold lines between cardinal regions, we define that an object is only described within the cardinal region w.r.t. a LCS of another if *most* of its occupancy region overlaps with that cardinal region.

4.2 Representation: from continuous data to symbolic relations

Symbolic descriptions of the scene, written using the syntax of prolog facts[3], are formulated at each key-frame. A key-frame occurs at the end of a motion event. This allows for a compact representation of objects and actions in the scene. For each key-frame, we note the following objects and relations:

- For each salient object, its existence and properties:
  - *object(obj1).*
  - *rel(property, o1, colour4)*, meaning that the object o1 has the property *colour4*.
- The displacement of one object which is placed on the cardinal position w.r.t. the local frame of reference of another object:
  - *rel(move, o2, ne, o1)*, meaning the object *o2* was moved to a position northeast (*ne*) w.r.t. *o1*.

  Assume that the symbols *lgray* and *dgray* refer to, respectively, *light gray* and *dark gray* (relating to the colours of the blocks in Figure 3), and a symbol *ne* representing the direction northeast. It is now possible to describe the sequence of objects shown in Figure 3 by the set of statements in Figure 4 below.

obj(o1).
rel(property, o1, lgray).
obj(o2).
rel(property, o2, dgray).
rel(move, o2, ne, o1).
obj(o3).
rel(property, o3, lgray).
rel(move, o3, ne, o2).

Figure 4: Symbolic description of Figure 3.

Such sets of statements are handled by an Inductive Logic Programming system that contributes with a model for the focus of attention that provides the expectation about what object should be placed in which position. This constitutes our spatial attention mechanism.

4.3 Reasoning: symbolic learning using Inductive Logic Programming

The spatial attention is learned using the Inductive Logic Programming system named Progol [Muggleton, 1995; 2001]. Progol works by generalising a set of positive only examples according to user-defined mode declarations. Mode declarations are a set of instructions on the general form of the data generalisation required. Informally, Progol allows a set of noisy positive examples to be generalised by inductively subsuming the data representations by more general data representations/rules (with the aim of reducing representational complexity, without over-generalising).

The aim of inductive learning in this work is two fold. First, it is to obtain a set of rules for deciding which block to move, and where to move it according to the pattern of objects observed. A second motivation is to use these rules to guide the focus of attention. We may wish to say:

- move block *obj16* to a spatial position;
- move a block with property *colour4* to a spatial position;
- move any block to a spatial position;
- pan and tilt the camera to a position where a particular object is expected to be placed.

From running Progol for a synthetic data set representing a similar situation to that in Figure 3, but containing 20 objects (positive examples), we obtained the rules in Figure 5.

rel(move,A,ne,B) :- rel(prop,A,lgray), rel(prop,B,dgray).
rel(move,A,ne,B) :- rel(prop,A,dgray), rel(prop,B,lgray).

Figure 5: Induced rules with respect to the example in Figure 3.

The rules in Figure 5 state that any object A should be moved to a position northeast of any object B if A is light (dark) gray and B is dark (light) gray. These learned rules can be used either to predict which object should be placed w.r.t. another object (and in which position) or to actually move an object to a position according to the pattern observed.

In order for an agent to use this learned guide-rules for focus of attention, it needs to convert back from a symbolic spatial description to a continuous pan-tilt angle description, this is done by choosing the appropriate position that would place the object at the cardinal position inferred and at a distance that is equal to the distance between the nearest object (that provided the former a reference frame) and the latter own referent object (that provided its frame of reference). Figure 6 illustrates the possible positions in which an object could be placed.

It is worth pointing out that Progol needed not fewer than 20 examples to learn a proper model of expectancy from synthetic data about this domain. The quality of the learned rules degraded gracefully with respect to a decrease in the number of examples available. A considerable increase in the size of the data sets may be needed if real data were used. An evaluation of our solution in this case is left for future investigations.

The next section introduces, by means of examples, some of the future experiments that are going to be used to evaluate the performance of our spatial attention mechanism from data provided by the vision system described in Section 2.

5 Further examples (or future experiments)

Figure 7 shows four settings where the system above is going to be tested. The first three arrangements (Figures 7(a),7(b)

[3]In the syntax of prolog constants are represented with lowercase characters while variable with upper case.

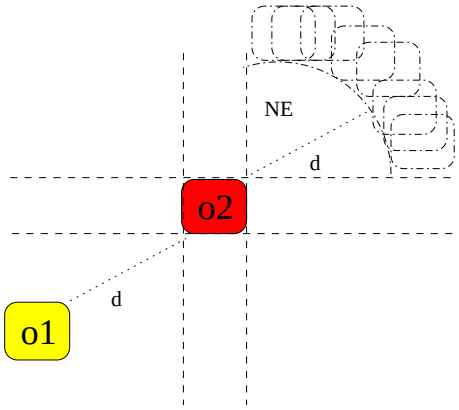
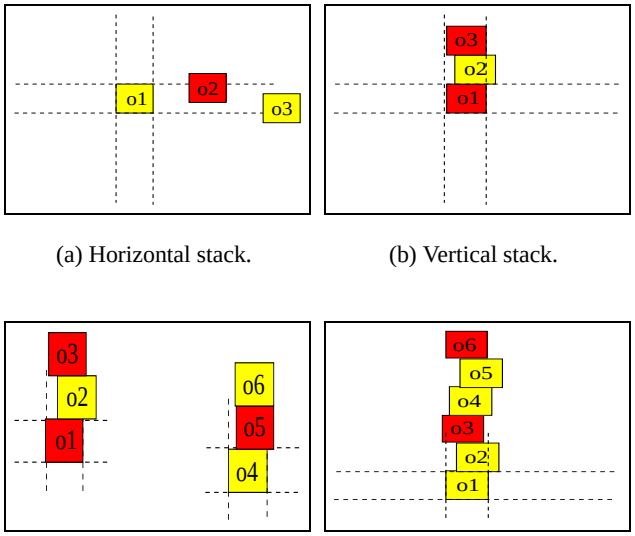


Figure 6: The location of possible positions in which to place objects given a spatial description, where 'd' is a distance value.



(a) Horizontal stack. (b) Vertical stack.

(c) Two towers. (d) Longer repetition sequence.

Figure 7: Examples.

and 7(c)) are just variations of the example depicted in Figure 3. Learning rules concerning the focus of attention in these cases are straightforward tasks, according to experiments with synthetic data. However, applying our solution to real data from many different spatial domains will allow us to evaluate the robustness of the entire system with respect to different spatial dimensions. Moreover, error in pan-tilt angles may generate different kinds of spurious formulae during the inductive learning process, warrantying further research in inductive logic programming. The example in Figure 7(d) may come as a challenge for Progol to learn, as we have encountered problems in inducing facts dependent on various previous states [Needham *et al.*, 2005]. This problem indicates roads for future investigations.

6 Discussion and future works

This paper discussed work in progress that aims the inductive learning of focus of attention from the visual observation of agents executing tasks in the world. The tasks assumed so far are the construction of repetitive patterns with coloured blocks, so that the system (after assimilating the pattern) could build a model of expectancy about where to place the objects in the pattern and which object should be placed.

In the present stage of this research an active vision system was developed that comprises a first level of attention, named *data driven visual attention*. On a second level, a neural network provides a statistical model of expectancy that should compete with a symbolic model learned by inductive logic programming.

In this work, we discussed some examples of how this system could be evaluated on real data, so far the system has only been used on synthetic data. Future work shall focus on this. However, due to our previous success on integrating computer vision with inductive logic programming for learning protocols from observation [Needham *et al.*, 2005][Santos *et al.*, 2004][Magee *et al.*, 2004], we are very confident that the framework proposed in this paper will provide a powerful solution for learning visual attention.

A key issue, left for future research, is how to scale the methodology presented in this work for learning visual attention from the observation of more complex actions than those discussed above.

7 Acknowledgements

This work was partially funded by the European Union, as part of the CogVis project.

References

[Baluja and Pomerleau, 1997] Shumeet Baluja and Dean Pomerleau. Dynamic relevance: Vision-based focus of attention using artificial neural networks. (technical note). *Artificial Intelligence*, 97(1-2):381–395, 1997.

[Bobick, 1997] A. Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical transactions of the royal society, London*, 352:1257–1265, 1997.

[Dickmanns, 1992] Ernst D. Dickmanns. Expectation-based dynamic scene understanding. In Blake and Yuille, editors, *Active vision*, pages 303–334. MIT Press, 1992.

[Fitzpatrick *et al.*, 2003] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning about objects through action - initial steps towards arfificial cognition. In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, pages 3140–3145, 2003.

[Johnson and Hogg, 1996] N. Johnson and D. C. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14:609–615, 1996.

[Magee *et al.*, 2004] D. Magee, C. Needham, A. Cohn P. Santos, and D. Hogg. Autonomous learning for a cognitive agent using continuous models and inductive logic programming for audio-visual input. In *Proceedings of the AAAI workshop on Anchoring Symbols to Sensor data*, 2004.

[Mann *et al.*, 1997] Richard Mann, Allan Jepson, and Jeffrey Mark Siskind. The computational perception of scene dynamics. *Computer Vision and Image Understanding: CVIU*, 65(2):113–128, 1997.

[Muggleton, 1995] S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995.

[Muggleton, 2001] S. H. Muggleton. Learning from positive data. *Machine Learning*, 2001.

[Needham *et al.*, 2005] C. Needham, P. Santos, D. Magee, V. Devin, D. Hogg, and A.G. Cohn. Protocols from perceptual observations. *Artificial Intelligence*, 2005. accepted, pending minor revisions.

[Santos *et al.*, 2004] P. Santos, D. Magee, A. Cohn, and D. Hogg. Combining multiple answers for learning mathematical structures from visual observation,. In *Proceedings of the European Conference on Artificial Intelligence (ECAI-04)*, Valencia, Sp, 2004.

[Shanahan, 2002] M.P. Shanahan. A logical account of perception incorporating feedback and expectation. In *Proceedings of KR 2002*, pages 3–13, 2002.

[Siskind, 1995] J.M. Siskind. Grounding language in perception. *Artificial Intelligence Review*, 8:371–391, 1995.

[Tsotsos, 1985] J.K. Tsotsos. The role of knowledge organization in representation and interpretation of time-varying data: The alven system. *Computational Intelligence*, 1(1):16–32, 1985.

[Tsotsos, 2001] J.K. Tsotsos. Motion understanding: task-directed attention and representations that link perception to action. *International Journal of Computer Vision*, 45(3):265–280, 2001.