INDUCTIVE LEARNING SPATIAL ATTENTION

Paulo Santos[*]
psantos@fei.edu.br

Chris Needham[†]
chrisn@comp.leeds.ac.uk

Derek Magee[†]
drm@comp.leeds.ac.uk

[*]Department of Electrical Engineering
Centro Universitário da FEI, São Paulo, Brazil

[†]School of Computing
Leeds University, Leeds, UK

ABSTRACT

This paper investigates the automatic induction of spatial attention from the visual observation of objects manipulated on a table top. In this work, space is represented in terms of a novel observer-object relative reference system, named Local Cardinal System, defined upon the local neighbourhood of objects on the table. We present results of applying the proposed methodology on five distinct scenarios involving the construction of spatial patterns of coloured blocks.

KEYWORDS: Qualitative Spatial Reasoning, Cognitive Vision

RESUMO

A proposta deste artigo é investigar a indução automática do foco de atenção a partir da manipulação de objetos sobre uma mesa. Neste trabalho espaço é representado em termos de uma nova proposta de um sistema de referências relativo ao observador e aos objetos. Este sistema de referências chama-se Sistema Cardinal Local e é definido sobre a vizinhança local dos objetos na mesa. Resultados da aplicação da metodologia proposta são apresentados a partir de cinco cenários envolvendo a construção de pilhas de blocos.

PALAVRAS-CHAVE: Raciocínio Espacial Qualitativo, Visão Cognitiva

1 INTRODUCTION

The development of a computer vision system capable of directing its focus of attention towards what has been perceived as most relevant in a dynamic scene, given the recent history of observations, is of essential importance in order to reduce the computational cost involved in image processing. However, as pointed out in (Tsotsos, 2001), the subject of task-directed attentive processing has been a theme largely neglected in computer vision and image understanding. Authors have been making strong assumptions about attention in order to develop other issues in computer vision, assumptions such as: a one-to-one correspondence between figures in adjacent frames (Siskind, 1995); or the *a priori* definition of regions of interest in the images that are manually given as inputs to the vision systems (Bobick, 1997). A few authors have proposed models for predicting where to search for corresponding regions from image to image (Shanahan, 2002)(Dickmanns, 1992)(Baluja and Pomerleau, 1997). However, the problem of how such expectancy models could themselves be automatically learned from the visual observation of tasks has only recently being addressed (Tsotsos et al., 2005)(Khadhouri and Demiris, 2005).

The present paper investigates the development of a knowledge-based system capable of automatically inducing

Artigo submetido em 10/12/2007
1a. Revisão em 28/02/2008
2a. Revisão em 09/05/2008
Aceito sob recomendação do Editor Associado
Prof. Anna Helena Reali Costa

the focus of attention from the visual observation of tasks being performed on a domain. We tested the methodology proposed on five distinct scenarios whereby the system was capable to infer a different, and appropriate, attention mechanism for each of the given tasks. These results suggest that the research reported here is eligible to be applied on learning suitable attention mechanisms from the observation of various distinct situations in a dynamic world. In contrast to the work described in (Tsotsos et al., 2005) and (Khadhouri and Demiris, 2005), which take a biologically inspired perspective, the present work experiments inducing attention from a symbolic perspective, whereby it is possible to make explicit issues about spatial knowledge representation.

Having its emphasis on inducing knowledge from visual observation, this work falls under the umbrella of cognitive vision whose main purpose is to develop computer vision systems capable of extracting knowledge about the environment observed, and infer new information from this knowledge. Within cognitive vision systems the research reported here follows the framework presented in (Needham et al., 2005) where a cognitive vision system capable of learning protocols from the visual information of dynamic scenes is proposed. In fact, the present paper is an extension of the work reported in (Magee et al., 2005); however, in that work we were interested in the autonomous learning of rules to control a vision system simulating saccadic eye movements, whereas here we concentrate on the process of learning spatial attention *per se*, providing a more complete set of experiments on this subject.

The present work assumes the observation of patterns in space formed by coloured blocks that are stacked by an agent in such a way to create repetitive sequences of colours. These patterns are input to an inductive logic programming (ILP) system (Mitchell, 1997) that is used to generate a *model of expectancy* about what should be the next object to be moved and in which position it should be placed. This provides the basis for a spatial attention mechanism with which an autonomous agent can predict the location and the nature of an event that is about to occur given the observed pattern. Therefore, the resulting model of spatial expectancy is learned from the observation of agents acting in the external world. Underlying the development of this project is our long term goal to induce spatial relations from observing the commonsense world, an issue that has so far only been glimpsed at in the literature (Kaelbling et al., 2001)(Cha and Gero, 1998).

In order to represent the domain objects, this paper defines an observer-object relative reference system named *Local Cardinal System* (LCS), whereby each new block that is moved in the observed situation is located according to a cardinal reference frame defined by the nearest object to this block. Local Cardinal System is introduced in Section 2. The domain objects are represented by a set of spatial relations introduced in Section 3. The symbolic learning system used for learning spatial attention in this domain is presented in Section 4. Section 5 discusses some results and Section 6 concludes this paper. Throughout this paper we use the Prolog syntax whereby variables are upper-case letters and constants, lower-case.

2 LOCAL CARDINAL SYSTEM

We assume a domain populated by 2D perspective projections from 3D convex objects, placed on a table top, observed by a video camera. For the purposes of this paper the domain objects are simply referred to as *objects*. In this domain, objects are located according to observer-object relative frames of reference, named Local Cardinal Systems (LCS). Figure 1 depicts three LCS defined with respect to the viewpoint of the reader of this paper.

Local Cardinal Systems work in the following way. Each object that is placed on the table defines its own cardinal reference frame which is used to locate other objects that lie in its *local neighbourhood*; i.e., each object is located with respect to the LCS of its nearest neighbour (or neighbours if it is the case that more than one neighbour dist the same amount to the object they are locating). The cardinal directions of each LCS are bounded by the extreme points of the referent object's boundaries and are dependent on the observer's viewpoint. For instance, the north and south regions of an object are bounded by two parallel lines, each of them passing through the left and right extreme points of this object and directed vertically with respect to the observer's viewpoint. In other words, the boundaries of the north (south) direction of a LCS will always be orthogonal to the gaze direction of an observer looking at the object, with a negative (positive)

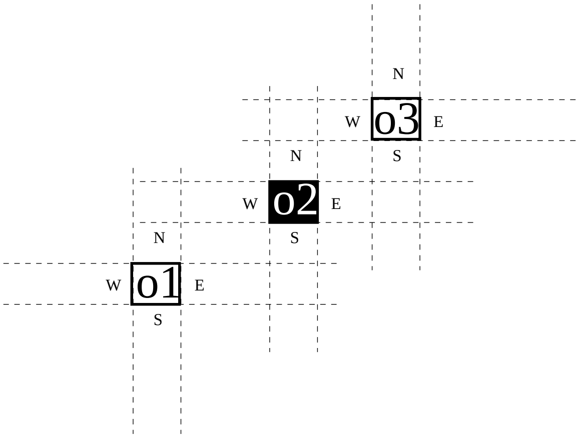

Figure 1: Local Cardinal Systems.

vector cross product with respect to this direction (cf. Figure 1). Analogously, the west and east regions of a LCS are bounded by two parallel lines (each of them passing through the top and bottom extreme points of the referent object) that are perpendicular to the lines defining the north and south regions. The northeast, southeast, northwest and southwest directions are defined accordingly.

Local Cardinal Systems assume also the following *precedence constraint*: an object is only described within the reference frame of another if the former is placed *after* the latter. For instance, objects already placed on the table are not described in the reference frames of newly placed ones. Thus, this constraint implies an implicit notion of temporal precedence in the way the objects are represented within LCS. This implicit temporal ordering could be used in future work to facilitate temporal reasoning within Local Cardinal Systems.

The assumptions of local-neighbourhood descriptions and precedence constraint facilitates an efficient qualitative description of the location of objects in space. In other words, locating one object (o) in the local neighbourhood of n others, respecting a precedence constraint, implies a running-time (and space) complexity of $O(n)$ in the worst case, whereby o is placed on the centre point of a circle of n objects. Another consequence of these assumptions is that local cardinal systems define an intransitive, asymmetric and irreflexive relation of location.

In order to exemplify how objects are located within Local Cardinal Systems, consider that in the situation depicted in Figure 1 Object **o3** was placed on the table after Object **o2**, and that the latter was placed after Object **o1**. Thus, according to our definition of LCS, **o3** is located on the north east of **o2**; however, the location of the latter cannot be related with respect to the former due to the precedence constraint. Moreover, **o2** is on the north east of **o1**, but **o3** cannot be described within the LCS of **o1** since **o1** is not the nearest object to **o3**.

In order to avoid ambiguous descriptions when an object falls on the threshold lines between cardinal regions, we assume that an object is only described within a particular cardinal region of a LCS if *most* of its occupancy region overlaps with that cardinal region. If a threshold line divides the object in halves, we assume that the northern-most or the western-most cardinal region dominates the object's location; we also assume that northern dominates western regions.

It is worth noting the similarity between Local Cardinal System and the double-cross calculus proposed in (Freksa, 1992)(Scivos and Nebel, 2001), whereby the qualitative information available for an observer in a 2D situation is expressed in terms of a set of 15 location relations obtained from the combination of front-back and left-right dichotomies. The double-cross calculus, however, is defined on the location of points and does not assume a precedence constraint as the LCS does. This constraint should imply that reasoning with LCS is computationally simpler than reasoning with the double-cross calculus since not all of the compositions between the double-cross relations are permitted in LCS. The investigation of the algorithmic properties of LCS, and how it relates to the double-cross calculus, however, is left for future work.

3 FROM CONTINUOUS DATA TO SYMBOLIC RELATIONS

For the experiments reported below we assume the vision setup composed of a video camera observing a table top where blocks are being stacked. Figure 2 shows a picture of this setup, and also depicts a schema of the system modules.

This work assumes that the data obtained by a vision system is turned into a symbolic description of states of the objects observed on the table top. This is used in turn as input data for the Inductive Logic Programming (ILP) module.

To turn video streams into symbolic information, the vision system uses motion as the cue to select interesting portions of the image stream, this amounts to the early attention module in Figure 2. Based on a generic blob tracker (Magee, 2004), this mechanism works on the principle of multi-modal background modelling and foreground pixel grouping. Thus, the bounding box, centroid location and pixel segmentation are extracted from any moving object in the scene in each frame of the video sequence. This mechanism then identifies key scenes where there is qualitatively no motion for a number of frames, which are preceded by a number of frames containing significant motion. For each object in the selected frames, its colour is represented as one of the 11 basic Berlin-

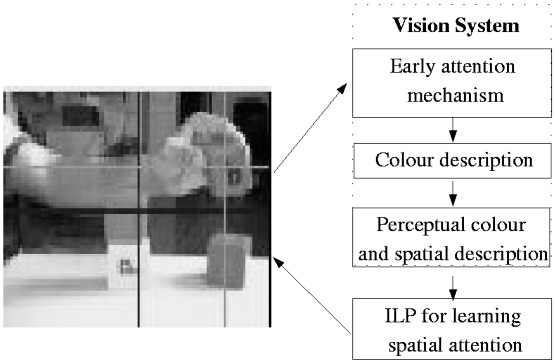
Vision System
Early attention mechanism
Colour description
Perceptual colour and spatial description
ILP for learning spatial attention

Figure 2: A scheme of the setup.

Kay colours (Berlin and Kay, 1969), initiating the *colour description module* in Figure 2. In this procedure the modal colour of foreground pixels associated with the object is extracted by building a histogram in Hue-Saturation space. The bin with the highest frequency is considered to be the modal bin. The modal colour is determined by selecting the example from this bin with the closest intensity to the mean intensity for this object. This is converted to a perceptual colour using the Consensus-Colour Conversion of Munsell colour space (CCCM) used in (Gilbert and Bowden, 2005). In the sequence, the positions of new objects in each selected scene are described in terms of Local Cardinal Systems introduced in Section 2.

The perceptual colour detection and the description of objects in terms of LCS completes the computer vision processing as schematised in Figure 2. This system facilitates the following representation:

- For each salient object, its existence and properties are represented by:
    - *object(o1).*
    - *rel(property, o1, colour4)*, meaning that the object $o1$ has the property *colour4*.
- The displacement of one object to a particular position with respect to the local frame of reference of another object is then represented as:
    - *rel(move, o2, ne, o1)*, meaning that the object *o2* was moved to a position northeast (*ne*) of *o1*

  assuming the symbols *white* and *black* for the colours of objects in Figure 1, and the symbol *ne* representing the direction northeast. Thus, the vision system presented above would describe the situation depicted in Figure 1 by the set of statements shown in Figure 3 below, for instance.

obj(o1).
rel(property, o1, white).
obj(o2).
rel(property, o2, black).
rel(move, o2, ne, o1).
obj(o3).
rel(property, o3, white).
rel(move, o3, ne, o2).

Figure 3: Symbolic description of Figure 1.

Sets of statements such as these are input to an inductive logic programming system that generates a model for spatial attention to the particular situation observed. This issue is discussed in the next section.

4 SYMBOLIC LEARNING USING ILP

In previous works (Santos and Shanahan, 2002; Santos and Shanahan, 2003; Santos, 2007; dos Santos et al., 2008) we have concentrated on developing systems capable of generating explanations for computer vision data using abductive reasoning. Abduction was proposed by Charles Peirce as *the inference that rules the first stage of scientific inquiries and of any interpretive process* (Peirce, 1958), i.e., the process of suggesting hypotheses to explain a given phenomenon. In Peirce's terms, an explanation of a phenomenon supplies a proposition which, if it had been known to be true before the phenomenon presented itself, would have rendered that phenomenon predictable. In this sense, abductive reasoning can be understood as the inverse of deductive reasoning, since abductive inference goes from data (observations) to explanatory hypotheses, while deduction provides the consequences of assumed facts. A third inference method, *induction* is proposed in (Peirce, 1958) to cope with the generalisation of facts. Therefore, the process of scientific inquiry, according to Peirce, is composed by three stages. First, abduction proposes explanations of observations, the consequences (or predictions) of these hypotheses are traced out by deduction which are, then, compared to results of experiments by induction. The hypothesis underlying the present work is whether this procedure could be applied to visual perception. In this context, the purpose of the present paper is the investigation of how inductive inference could be used to learn an agent's focus of attention. The integration of these various inference patterns into a single intelligent system is left for future work.

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 a visual agent's focus of attention. Thus, we may wish to say:

- move block *o16* to a spatial position;
- move a block with property *colour4* to a spatial position;
- move any block to a spatial position;
- create an expectancy about which position a particular object is likely to be placed.

In the present paper, spatial attention is learned using the inductive logic programming system named Progol (Muggleton and Firth, 2001)(Muggleton, 1995)(Muggleton, 1996), which generates a logic program that generalises a set of positive-only examples. Progol's capability of inducing rules from datasets containing solely positive examples is the feature that makes it suitable for the task of learning rules from passive observations, where negative examples are not available.

The expectancy model that guides the focus of attention is obtained by Progol as follows. The task is to induce a logic program H, which combined with a set of statements $obj(Obj)$ and $rel(property, Obj, Colour)$ (such as those included in Figure 3), composing a background theory B, entails the observations of motion of objects on the table top, represented by a set of atoms $rel(move, Obj_i, Position, Obj_j)$ (the set of examples E). Formally, given E and B the task of induction is to find H such that:

$$B, H \models E.$$

Taking for instance the situation described in Figure 3 as input to Progol, and including mode declarations that limit the search to rules with $rel(move, A, N, B)$ as head and any number of $rel(property, A, Colour)$ in their bodies, Progol induced Rules (1) and (2) below. These rules represent that *any object A should be moved to a position northeast of any object B if A is white and B is black* (cf. Rule (1)); and, conversely, that *any object A should be moved to a position northeast of any object B if A is black and B is white* (cf. Rule (2)). This pair of rules compose an expectancy model for the scene observed.

$$rel(move, A, ne, B) :- \quad (1)$$
$$rel(property, A, white), rel(property, B, black).$$
$$rel(move, A, ne, B) :- \quad (2)$$
$$rel(property, A, black), rel(property, B, white).$$

More precisely, Progol is given a knowledge base containing a sequence of formulae such as that presented in Figure 3 (descriptions of the visual data) and the mode declarations shown in Formulae (3) and (4) below.

$$:-modeh(rel(\#reltype, +obj, \#loc, +obj)) \quad (3)$$
$$:-modeb(rel(\#reltype, -obj, \#property)) \quad (4)$$

In Formulae (3) and (4) $modeh$ and $modeb$ force Progol to find Horn clauses having $rel(\#reltype, +obj, \#loc, +obj)$ in the head and $rel(\#reltype, -obj, \#prop)$ in the body (respectively), whereas $reltype$ is a variable for $property$ or $move$, loc is a direction of the LCS and obj is a variable for an object. The symbols $\#$, $-$ and $+$ sets Progol to search, respectively, for a constant, an output and an input variable for the variable type to which it is attached.

From these mode declarations, Progol generates initially the following most specific clause (for brevity we abbreviate the term $property$ by $prop$ in the formulae below):

$$rel(move, A, ne, B) :-$$
$$rel(prop, A, white), rel(prop, B, black),$$
$$rel(prop, C, white), rel(prop, D, black),$$
$$rel(prop, E, black), rel(prop, F, white),$$
$$rel(prop, G, black), rel(prop, H, white),$$
$$rel(prop, I, black), rel(prop, J, black),$$
$$rel(prop, K, white), rel(prop, L, black),$$
$$rel(prop, M, white), rel(prop, N, black),$$
$$rel(prop, O, black), rel(prop, P, white),$$
$$rel(prop, Q, black), rel(prop, R, white),$$
$$rel(prop, S, black), rel(prop, T, black).$$

which is reduced to the rule:

$$rel(move, A, ne, B) :- \quad (5)$$
$$rel(prop, A, white), rel(prop, B, black).$$

by just reducing redundancies. The system then generates all possible generalisations of this rule as shown below.

$$rel(move, A, ne, B).$$
$$rel(move, A, ne, B) :- rel(prop, A, white).$$
$$rel(move, A, ne, B) :- rel(prop, C, white).$$
$$rel(move, A, ne, B) :- rel(prop, B, black).$$
$$rel(move, A, ne, B) :- rel(prop, C, black).$$
$$rel(move, A, ne, B) :- rel(prop, A, white),$$
$$rel(prop, B, black).$$
$$rel(move, A, ne, B) :- rel(prop, A, white),$$
$$rel(prop, C, black).$$
$$rel(move, A, ne, B) :- rel(prop, C, white),$$
$$rel(prop, B, black).$$
$$rel(move, A, ne, B) :- rel(prop, C, white),$$
$$rel(prop, D, black).$$

These generalisations are evaluated by their coverage on the entire example set (cf. (Muggleton and Firth, 2001)) and the best evaluated rules are output. In the present case these are the Rules (1) and (2) above.

The model of expectancy represented by Rules (1) and (2) tells us about which object is deemed to be placed on the table (and where) with respect to another object already resting on the table top. Expectancy in this context would be represented by a query such as $rel(move, A, Position, b)$ read as “which object A should be moved to a position $Position$ with respect to a particular object b ?”. If it is the case that b is (for instance) a black object, the only possible instantiation for the query variables that renders it true (given Rules (1) and (2)) is the unification of $Position$ with the constant ne and A with a white object in the domain (from Rule (1)). Consequently, if the body of the rules are interpreted as preconditions to the application of the action represented in the

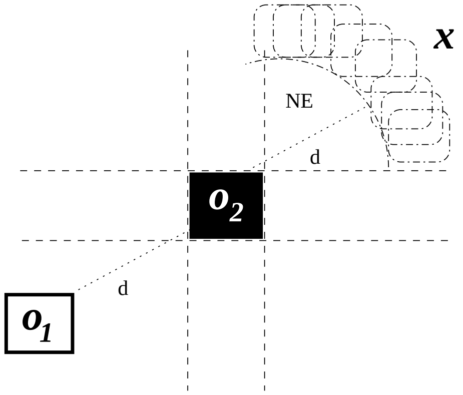

Figure 4: The location of possible positions in which to place an object x given a spatial description, where 'd' is a distance value.

rule heads, the rules above could be used by an agent to actually move an object to a particular position, according to the pattern observed.

In practice, in order to an agent use these 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 (for instance) the appropriate position that would place an object x (see Figure 4) at the cardinal position inferred by the learned rules and at a distance that is equal to the distance between the nearest object o_2 (that provided x a reference frame) and the object that provided a frame of reference to o_2.

It is worth pointing out that Progol needed 20 (noise-free) examples to learn the rules above. The quality of the learned rules degraded gracefully with respect to a decrease in the number of examples available.

The next section discusses some results obtained by running Progol to induce spatial attention from five distinct spatial arrangements of blocks.

5 EXPERIMENTS

In this section we discuss the results of applying the framework proposed above to generate rules for spatial attention from the observation of five distinct scenarios, shown in Figure 5.

In the first experiment (Figure 5(a)) the ILP system was input with data representing a stack of simply alternating white and black blocks: a white block is always stacked on top of a black block and *vice-versa*. The second experiment extends the previous test by assuming two stacks of simply alternating coloured blocks, instead of just one stack (Figure 5(b)), the idea here is to verify whether the system could abstract away the tag naming the stacks finding a single set of rules for both piles of blocks. In contrast, in the third experiment (Figure 5(c)) the system has to handle longer sequences of coloured blocks, whereby two white blocks are always followed by one black block (and on this block two white blocks are stacked). The case shown in Figure 5(c) represents a greater level of complexity when compared with the previous ones, as more rules are necessary in order to account for the observed pattern. The fourth experiment increases the complexity of the previous case by assuming two stacks with distinct patterns: one stack has two white blocks followed by one black, and the other has two black blocks followed by one white (cf. Figure 5(d)). Therefore, in contrast to the second experiment, we expect that the induced rules make a distinction about which stack they refer to. The fifth and final experiment was designed to test whether the system could generate a set of rules representing that the most distinctive feature in the scenario was the *direction* induced by the block colours (and not the sequence of colours). In this scenario, a block is always placed on top of white or grey blocks, whereas it always goes to the east of a black block. Figure 5(e) depicts three distinct stacks constructed according to these constraints.

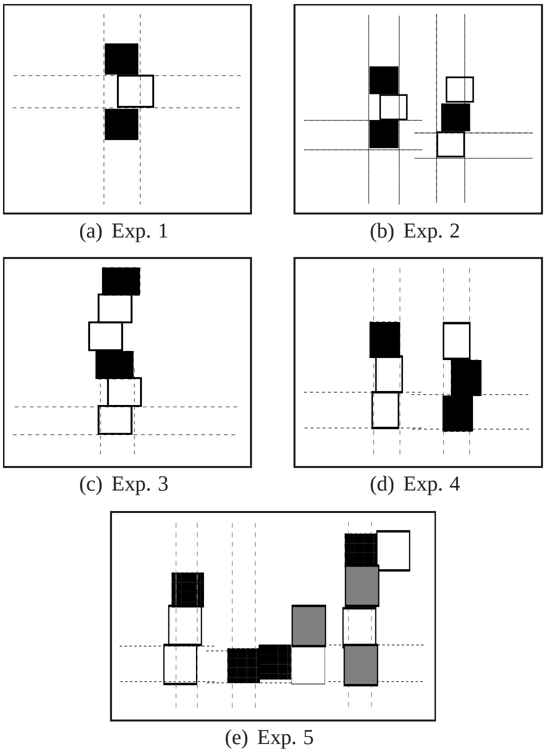
(a) Exp. 1
(b) Exp. 2
(c) Exp. 3
(d) Exp. 4
(e) Exp. 5

Figure 5: Experiments.

Exp.1 - Simply alternating stack

From a dataset obtained by the vision system (described in Section 3) observing situations such as that depicted in Figure 5(a), the system induced the following two rules:

$$rel(move, A, n, B) :- \quad (6)$$
$$rel(property, A, white), rel(property, B, black).$$
$$rel(move, A, n, B) :- \quad (7)$$
$$rel(property, A, black), rel(property, B, white).$$

which are, essentially, the rules used as an example in Section 3 above. For clarity, Rule (6) represents that "every block A should be moved to the north of a block B if A is white and B is black" and that "every block A should be moved to the north of a block B if A is black and B is white" (Rule (7)).

The data from which the rules above were generated was obtained in seven sections, representing the observation of the construction of seven simply alternating stacks composed of four blocks each. We used blocks coloured in blue or green in order to obtain great accuracy from the colour classification module[1]. The seven data sets were merged into a single one to provide enough data redundancies so that an appropriate model could be generalised. Thus, the appropriate rules for the focus of attention were induced from 28 data items obtained from the vision system. Along with Rules (6) and (7) above, Progol also singled out in its answer set two rules: $rel(move, p1, nw, p0)$ and $rel(move, p2, w, p0)$, where $p0$, $p1$ and $p2$ are particular grounded instances of objects in the training data (contrasting with the variables A, B representing ungrounded variables for objects in Rules (6) and (7)). These two atomic rules represent noise in the training data, where the relative positions of the objects $p0$, $p1$ and $p2$ in a stack have been mis-described by the tracking module, not agreeing therefore with the generalised Rules (6) and (7). It is worth noting that, in this case, the noisy rules were specific instances of the relation $rel/4$ for particular constants of the domain, therefore, the large majority of the domain objects will be handled by Rules (6) and (7) which are precise descriptions of the domain observed. This exemplifies Progol's robustness to noise in the vision data from simple scenarios (Needham et al., 2005)(Santos et al., 2004).

Exp.2 - Two stacks simply alternating

In this experiment the training set input to Progol describes the observation of two stacks of simply alternating coloured blocks, conform depicted in Figure 5(b). Each of these stacks was represented by the symbols t_1 and t_2. Thus, the predicates representing the motion events in the dataset for Exp. 2 had their argument extended to include the symbol specifying which stack a particular block is placed on. In practice, in the data file for Exp.2 the motion event was represented by

$$rel(move, \tau, < block_i >, < direction >, < block_j >),$$

where τ is either t_1 or t_2.

Analogously to Exp.1, the system obtained Rules (8) and (9) which appropriately represent the pattern of the motion event observed from the data set representing the two stacks in Figure 5(b). In these rules the system correctly abstracted as irrelevant the fact about the existence of two distinct stacks, as the pattern of blocks in both stacks was exactly the same. This fact is represented by the variable A in the rules below.

$$rel(move, A, B, n, C) :- \quad (8)$$
$$rel(property, B, white), rel(property, C, black).$$
$$rel(move, A, B, n, C) :- \quad (9)$$
$$rel(property, B, black), rel(property, C, white).$$

Exp.3 - Single stack, longer repetition sequence

In the third experiment (Figure 5(c)) we want to verify whether an appropriate expectancy model can be induced from the observation of a stack of two white blocks followed by one black block. The results obtained are Formulae (10)–(13) below.

$$rel(move, A, n, B) :- \quad (10)$$
$$rel(move, B, n, C), rel(property, A, black),$$
$$rel(property, B, white), rel(property, C, white).$$
$$rel(move, A, n, B) :- \quad (11)$$
$$rel(move, B, n, C), rel(property, A, white),$$
$$rel(property, B, black).$$
$$rel(move, A, n, B) :- \quad (12)$$
$$rel(move, B, n, C), rel(property, A, white),$$
$$rel(property, C, black), rel(move, C, n, A).$$
$$rel(move, A, n, B) :- \quad (13)$$
$$rel(move, B, n, A), rel(property, A, white).$$

The first rule found by Progol (Rule (10)) captures the main structure of the stack, i.e., a black block is always placed to the north of two white blocks. Rule (10) can be read as "every block A is moved to the north of any block B if A is black, B is white and there is a white block C that is placed on the north of B".

Rule (11) represents the fact that on the north of a black block always comes a white block. Progol also induced within this

[1] In this text we used white instead of blue and black in the place of green in order to make the figures clear in black&white printouts.

rule that black objects B lie on the north of some object C, but the property of the latter was not specified. The lack of information about the property of C is probably due to the fact that the number of black objects on the dataset is half the number of white objects. Therefore, any inductive hypotheses including this property was probably deemed as statistically irrelevant by the ILP system.

Progol also found two formulae representing some sort of “Escherian” stack (Formulae (12) and (13)), which represents that an object is placed to the north of another if the latter is also placed to the north of the former. This is probably due to the fact that Progol tries to generalise rules that explain each of the examples in the dataset, and not finite subsets of it. In contrast, in the present experiment, the important pattern occurs in sequences of three examples.

Formulae (10)–(13) were obtained from synthetic data representing a single stack of 56 blocks, fewer examples did not include enough redundancies to allow the induction of the above rules.

We also experimented on this scenario with a set of data obtained from the vision system described in Section 3. In this case, due to the limited camera view, the data provided to Progol was a combined set containing ten datasets, each of which composed by five-block stacks. Therefore, there were no connection between the top block of one stack and the bottom block of a following stack in the dataset, this implies a 20% loss of information in contrast to the synthetic dataset. Moreover, the vision system mis-described 10% of the colour or position of blocks in the data. The rules obtained in this case are Formulae (14)–(16) below. Even in this noisy situation, the system was capable of inferring Rule (14), which is analogous to Rule (11) above. However, the other two rules induced are not true with respect to the domain observed and Rule (14) alone does not provide a correct notion of the focus of attention within this domain. The problem here is that, due to limited information and noise in the data, the system needs a much greater number of examples to induce appropriate rules. The task of obtaining such a large dataset is not only tedious, but also jeopardises the use of the proposed system in more realistic scenarios. Possible avenues to cope with this issue are within our current research interests, and are discussed in the sequence.

$$rel(move, A, n, B) :- \quad rel(property, A, white), rel(property, B, black). \tag{14}$$

$$rel(move, A, n, B) :- \quad rel(property, A, white), rel(move, A, n, C), rel(move, C, n, A). \tag{15}$$

$$rel(move, A, n, B) :- \quad rel(move, B, n, A), rel(property, A, black). \tag{16}$$

In the following scenarios we only discuss the results obtained from synthetic data. The experiments with real data suffer from the same problem as reported in this subsection.

Exp.4 - Two stacks, longer repetition sequence

In this experiment the task is to induce two different patterns of colours from two distinct stacks (as shown in Figure 5(d)). The first is constructed using one black block after every sequence of two white blocks; while the second stack is the negative copy of the first, constructed with two black blocks followed by one white block. Rule (17) implies that every white block is placed on top of another white block in stack $t1$. Rules (18) and (19) encode the structure of stack $t1$ that was not captured by Rule (17); i.e., that black blocks are stacked on top of white blocks (Rule (18)) and that, to the north of black blocks, white blocks are stacked (Rule (19)). Rules analogous to (17) – (19) were also induced for stack $t2$, with their bodies representing the appropriate pattern of colours in this stack, as shown in Rules (20)–(22).

$$rel(move, t1, A, n, B) :- \quad rel(property, A, white), rel(property, B, white). \tag{17}$$

$$rel(move, t1, A, n, B) :- \quad rel(property, A, black), rel(property, B, white). \tag{18}$$

$$rel(move, t1, A, n, B) :- \quad rel(property, A, white), rel(property, B, black). \tag{19}$$

$$rel(move, t2, A, n, B) :- \quad rel(property, A, black), rel(property, B, black). \tag{20}$$

$$rel(move, t2, A, n, B) :- \quad rel(property, A, white), rel(property, B, black). \tag{21}$$

$$rel(move, t2, A, n, B) :- \quad rel(property, A, black), rel(property, B, white). \tag{22}$$

Formulae (17) – (19) are important facts about the domain observed. However, the rules obtained do not provide all the constraints regarding the structure of the stacks observed. More specifically, if Rules (17)–(19) were to be followed by an agent whose task is to reproduce the construction of the stacks observed, a stack composed of white blocks (with or without the occurrence of occasional single black blocks) would satisfy the agent’s knowledge about the domain observed.

Even providing an underconstrained model of the observed situations, the rules above provide an appropriate model of spatial attention in this domain as, recalling that Progol outputs the rules in decreasing order of statistical significance, it is apposite to make an initial hypothesis about the domain that to the north of every white block in Stack $t1$ comes another white block, according to Rule (17) (the first rule output by the system). The cases where this rule is not satisfied, i.e. the cases when a *black* block is stacked on $t1$ for instance, are represented by Rules (18) and (19) covering, thus, the pattern of colours in this domain.

Exp.5 - Change of direction

Figure 5(e) depicts the fifth experiment whose goal is to induce rules from spatial arrangements of three coloured blocks (black, white and grey) where black blocks in a stack force the next block in the sequence to be placed on its east side. Also, whenever any other block occurs, the next block is placed on the north of it.

In this experiment, the system induced Rules (23), (24) and (25) that appropriately capture the structure of the observed domain. Rule (23) represents the fact that any block A goes to the east of any black block B. Rules (24) and (25) express that any block A is moved to the north of any block that is either white or grey.

$$rel(move, A, e, B) :- rel(property, B, black). \tag{23}$$

$$rel(move, A, n, B) :- rel(property, B, white). \tag{24}$$

$$rel(move, A, n, B) :- rel(property, B, grey). \tag{25}$$

Therefore, whenever a black box B is observed, Rule (23) can be used to predict that an object A should be moved to the east of B. Similarly, Rules (24) and (25) allows an agent to predict that any object A should be moved to the north of an object B if the latter is either white or grey.

Discussion

The experiments discussed above indicate that the framework presented in this paper, which incorporates a novel qualitative reference system (Local Cardinal System) and inductive logic programming for learning spatial attention, provides appropriate models to hypothesise about where a synthetic agent should expect a particular object to be moved to, according to patterns observed in space. We investigated this framework conducting both, experiments on synthetic data and experiments on real data provided by a computer vision system.

From the experiments on synthetic data we verified that our framework was capable of inducing appropriate rules for spatial attention from a variety of (increasingly more complex) scenarios. We then conducted some experiments using data obtained directly from a video camera. In these cases the colour and position classification module of the vision system obtained an accuracy rate of 90%. The limited view of the video camera implies that only stacks with a small number of objects (generally with at most five blocks) can be observed. From such data, Progol was capable of learning analogous rules to those generalised from synthetic data with respect to the experiments involving colour patterns that could be described by two rules. For more complex patterns, the physical limitations of the vision system adds an extra level of complexity to the induction of rules from the domain. Consequently, a much greater number of examples (generally exponentially more) are needed in these cases than in synthetic descriptions of these domains. Possible solutions to this gap between the application of our proposed framework on synthetic and real domains are two fold. First, we need to improve the vision system by the use a pan-tilt camera widening the observer's field of view. This would allow the observation of arbitrary long patterns of coloured blocks. The colour and position classification modules should also be enhanced in order to provide cleaner data. Second, we should invest in more powerful inductive machinery that are not only capable of generalising from datasets, but also to hypothesise on possible (not necessarily generalising) hypotheses, either in an abductive fashion (Tamaddoni-Nezhada et al., 2006) or by descriptive mechanisms (Colton, 2002)(Santos et al., 2006). These should be fruitful venues for future work.

The expectancy models obtained by the inductive logic procedure described in this paper basically represent the spatial pattern of stack construction, giving no information about the time rate of the construction or about the ordering in which the stacks are built (in the case of scenarios with several stacks, such as those shown in Figures 5(b), 5(d) and 5(e)).We can say that the expectancy model learned, in the case of multiple stacks, assumes that the all stacks have the same probability of being the next to be modified. Research on learning probabilistic formulae that could provide an answer to this issue is well under way (Bennett and Magee, 2007).

We defined Local Cardinal Systems as convenient tools for describing the scenarios used throughout this paper. However, we believe that the investigation of LCS justifies a work by itself as they provide a rich (and efficient) formalism for locating objects in space from a commonsense representation perspective. Moreover, the implicit temporal dependency represented by the precedence constraint imposed on LCS (cf. Section 2) should be explored in further investigations in order to allow time-dependent rules to be induced, a challenging task for ILP systems (Needham et al., 2005).

6 CONCLUSION

This paper investigates the inductive learning of spatial attention from the visual observation of tasks being executed in the world. In this work we use data from a vision system, with a colour classification module, that provides a description of the observed scenes in terms of a novel cardinal reference system, named Local Cardinal System (LCS). In LCS each object that is moved to a location in the world defines its own cardinal reference frame which is used to locate other objects that are further moved to its local neighbourhood. The cardinal directions of local cardinal systems are defined according to the direction of the observer's gaze. Therefore, LCS is both observer and object relative. Moreover, LCS assumes a precedence constraint forcing objects already resting at particular locations not to be described in the reference frame of newly placed ones. This provides a simple and efficient way of representing the location of objects in space from a commonsense perspective. Data from the vision system, represented in terms of LCS, are input to the inductive logic programming system Progol whose task is to induce (for each observed spatial situations) rules representing expectancy models. These models include the location and the property of the next object to be moved in the observed domain. Therefore, an artificial agent could use the induced rules to either guide its gaze to the point in space where the next important event in the scene is expected to happen, or to actually execute actions whose effects would obey the protocols in the situation observed.

From the results discussed in this paper, and according to the previous success on integrating computer vision with inductive logic programming (Needham et al., 2005), we are very confident that the framework proposed in this paper provides a powerful tool for learning spatial attention from various distinct domains. How to scale the methodology presented in this work for learning visual attention from the observation of arbitrary tasks in the world is a challenging open issue for future research.

REFERENCES

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

Bennett, A. and Magee, D. (2007). Learning sets of submodels for spatio-temporal prediction, *Proceedings of the 27th Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence*, pp. 123–136.

Berlin, B. and Kay, P. (1969). *Basic Color Terms: Their Universality and Evolution*, University of California Press.

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

Cha, M. and Gero, J. (1998). Shape pattern recognition using a computable shape pattern representation, *Proc. of Artificial Intelligence in Design*, Kluwer, Dordrecht, Netherlands, pp. 169 – 188.

Colton, S. (2002). *Automated Theory Formation in Pure Mathematics*, Springer.

Dickmanns, E. D. (1992). Expectation-based dynamic scene understanding, *in* Blake and Yuille (eds), *Active vision*, MIT Press, pp. 303–334.

dos Santos, M., de Brito, R. C., Park, H.-H. and Santos, P. (2008). Logic-based interpretation of geometrically observable changes occurring in dynamic scenes, *Applied Intelligence* **1**.

Freksa, C. (1992). Using orientation information for qualitative spatial reasoning, *Theories and Methods of Spatial-Temporal Reasoning in Geographic Space*, Vol. 629 of *LNCS*, Springer-Verlag.

Gilbert, A. and Bowden, R. (2005). Incremental modelling of the posterior distribution of objects for inter and intra camera tracking, *Proc. of BMVC05*, Vol. 1, pp. 419–428.

Kaelbling, L., Oates, T., Hernandez, N. and Finney, S. (2001). Learning in worlds with objects, *Working Notes of the AAAI Stanford Spring Symposium on Learning Grounded Representations*, pp. 31 – 36.

Khadhouri, B. and Demiris, Y. (2005). Compound effects of top-down and bottom-up influences on visual attention during action recognition, *Proc. of IJCAI*, pp. 1458–1463.

Magee, D., Needham, C., Santos, P. and Rao, S. (2005). Inducing the focus of attention by observing patterns in space, *IJCAI Workshop on Modelling Others from Observations (MOO 2005)*, pp. 47 – 52.

Magee, D. R. (2004). Tracking multiple vehicles using foreground, background and motion models, *Image and Vision Computing* **20(8)**: 581–594.

Mitchell, T. M. (1997). *Machine Learning*, McGraw Hill.

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

Muggleton, S. (1996). Learning from positive data, *in* S. Muggleton (ed.), *ILP96*, Vol. 1314 of *Lecture Notes of Artificial Intelligence*, SV, pp. 358–376.

Muggleton, S. and Firth, J. (2001). CProgol4.4: a tutorial introduction., *in* S. Dzeroski and N. Lavrac (eds), *Relational Data Mining*, Springer-Verlag, pp. 160–188.

Needham, C., Santos, P., Magee, D., Devin, V., Hogg, D. and Cohn, A. (2005). Protocols from perceptual observations, *Artificial Intelligence Journal* **167**: 103–136.

Peirce, C. S. (1958). *Collected papers*, Harvard University Press.

Santos, P., Colton, S. and Magee, D. (2006). Predictive and descriptive approaches to learning game rules from vision data, *Proc. of IBERAMIA*, Vol. 4140 of *LNAI*, pp. 349–359.

Santos, P. E. (2007). Reasoning about depth and motion from an observer's viewpoint, *Spatial Cognition and Computation* **7**(2): 133–178.

Santos, P., Magee, D., Cohn, A. and Hogg, D. (2004). Combining multiple answers for learning mathematical structures from visual observation, *in* R. L. de Mataras and L. Saita (eds), *Proc. of ECAI*, IOS Press, Valencia, Spain, pp. 544–548.

Santos, P. and Shanahan, M. (2002). Hypothesising object relations from image transitions, *in* F. van Harmelen (ed.), *Proc. of ECAI*, Lyon, France, pp. 292–296.

Santos, P. and Shanahan, M. (2003). A logic-based algorithm for image sequence interpretation and anchoring, *Proc. of IJCAI*, Acapulco, Mexico, pp. 1408–1410.

Scivos, A. and Nebel, B. (2001). Double-crossing: decidability and computational complexity of a qualitative calculus for navigation, *in* D. Montello (ed.), *Spatial Information Theory: Foundations of GIS*, Vol. 2205 of *LNCS*, Springer, pp. 431 – 446.

Shanahan, M. (2002). A logical account of perception incorporating feedback and expectation, *Proc. of KR*, Toulouse, France, pp. 3–13.

Siskind, J. (1995). Grounding language in perception, *Artificial Intelligence Review* **8**: 371–391.

Tamaddoni-Nezhada, A., Chaleil, R., Kakas, A. and Muggleton, S. (2006). Application of abductive ILP to learning metabolic network inhibition from temporal data, *Machine Learning* pp. 209–230.

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

Tsotsos, J., Liu, Y., Martinez-Trujillo, J., Pomplun, M., Simine, E. and Zhou, K. (2005). Attending to visual motion, *Computer Vision and Image Understanding* **100**: 4–30.