Spatial Inference with Constraints

Christoph Schmoeger and **Carsten Gips** and **Fritz Wysotzki**
Berlin University of Technology
{schmohl, cagi, wysotzki}@cs.tu-berlin.de

Abstract

We present an approach for solving constraint nets occurring in spatial inference using methods of Machine Learning.

In contrast to qualitative spatial reasoning we use a metric description. Relations between pairs of objects are represented by parameterised homogeneous transformation matrices and numerical (nonlinear) constraints on the parameters. For drawing inferences we have to multiply the matrices and to propagate the constraints. The resulting constraint net consists of equations and inequalities containing trigonometric functions, which can be solved analytically only in rare cases. So we employ decision tree learning for learning and solving the constraints. We also use the decision trees for giving additional constraints for inferring a spatial relation from a set of other relations.

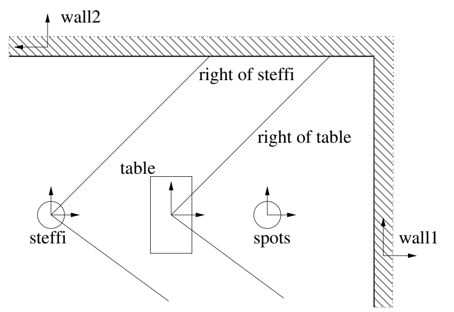
1 Introduction

The ability to draw conclusions is one of the most essential cognitive skills of humans. Especially in the context of spatial reasoning it is essential, not only for humans but also for mobile agents, to autonomously acquire additional information from a given spatial description, i.e. to infer relations not explicitly mentioned in the given description. This task is related to construct appropriate cognitive maps from a text describing a spatial configuration.

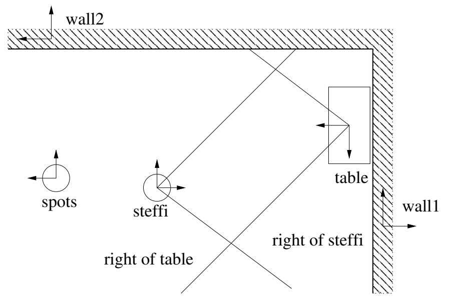
Given the example description (Fig. 1), the spatial relations `right(steffi, table)` and `right(table, spots)` can be extracted from the text. Since the description does not yield any information about the size and orientation of the described objects, there are several corresponding cognitive maps. In Fig. 2 two possible depictions, derived from the given facts, are shown[1].

> *Steffi is entering the room to feed her cat 'Spots'. She is looking around. To her right is a table. After looking further she finds her cat standing to the right of the table.*

Figure 1: Example text describing a spatial scenario



(a) Spots is to the right of Steffi



(b) Spots is to the left of Steffi

Figure 2: Appropriate cognitive maps to the example scenario

We use a metric approach for the inference procedure ([Wiebrock *et al.*, 2000]) instead of qualitative approaches to spatial reasoning ([Cohn and Hazarika, 2001]). The semantics of the spatial relations is given by homogeneous transformation matrices and constraints on the variables. With each object we associate a coordinate system, which is anchored in the centre of the object so that the x-axis is directed to the right and the y-axis points to the front of the object. As shown in [Wiebrock *et al.*, 2000], inference of a relation between two objects is done by propagating the constraints. Note, that we use an intrinsic interpretation of the spatial relations, i.e. the first object in the relation is the reference system for the second object. So we have to change the reference system while propagating constraints. This is done by multiplying the transformation matrices. Thus the resulting constraints contain trigonometric functions and inequalities and we can solve them analytically only in rare cases. Instead we apply methods of Machine Learning to spatial inference.

As shown in Fig. 2, there are many different interpre-

[1] Imagine the table as a desk, so it has an intrinsic front side.

tations of a given spatial description[2]. These maps may be even contradictory to each other. This is due to the intrinsic interpretation of the relations. Every conclusion depends on the relative position and orientation of the coordinate systems and objects, resp. In order to draw inferences we will usually need to give some additional constraints. These additional constraints are added to the premise, i.e. the given spatial description, and ensure the validity of the conclusion (cf. Sect. 3).

In this paper we show how to find these constraints using methods of Machine Learning, particularly an algorithm using CAL5-decision trees presented in [Gips *et al.*, 2002].

First we introduce spatial descriptions and the semantics of the spatial relations in Sect. 2. Thereafter we sketch the inference mechanism and introduce methods of Machine Learning for finding possibly necessary additional constraints for inference. In Sect. 4 we present results of our implemented inference algorithm and compare them with theoretical predictions. Finally, we state conclusions (Sect. 5).

2 Semantics of the Spatial Relations

As already mentioned, in contrast to qualitative techniques for spatial reasoning ([Cohn and Hazarika, 2001]), we use a metric approach known from the area of robotics.

With every object we associate a coordinate system, which is centered in the object, so that the x-axis corresponds to the right-direction and the y-axis to the front, and its shape and size. Spatial relations between pairs of objects are represented by their transformation matrices and constraints on the parameter of the matrices, i.e. restrictions to the relative positions and rotations of the objects.

So far we use projective relations like `right`, `left`, `front`, and `behind`. Let us consider the relation `right` in detail. The relation `right(table, spots)` places the cat 'Spots' right wrt. the table. The coordinate system, associated with the table, is the origin of the relation. The cat can be placed between the bisectors of the right angles of the table ((2) and (3) in Fig. 3). This is shown in Fig. 3, where Spots is represented by a circle and the table by a rectangle.

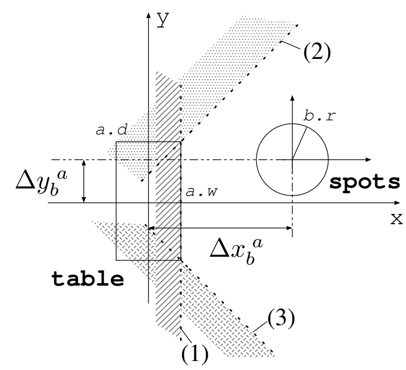


Figure 3: The relation `right(table, spots)` in detail

We can describe the relation `right(`a`, `b`)` mathematically by the following inequalities, i.e. constraints, whereby a stands for the table and b for the cat in the above example:

$$\Delta x_b^{\ a} \geq a.w + b.r \quad (1)$$

$$\Delta x_b^{\ a} \geq \Delta y_b^{\ a} + a.w - a.d + \sqrt{2}\, b.r \quad (2)$$

$$\Delta x_b^{\ a} \geq -\Delta y_b^{\ a} + a.w - a.d + \sqrt{2}\, b.r. \quad (3)$$

At this, $a.w$ and $a.d$ represent (the half of) the width and the depth of the rectangle, i.e. the table, and $b.r$ stands for the radius of a cylindric abstraction of the cat. The parameters $\Delta x_b^{\ a}$ and $\Delta y_b^{\ a}$ describe, as explained above, the relative position of object b wrt. object a. The constraints 1, 2, and 3 describe the corresponding half planes shown in Fig. 3. Generally the lower index denotes the object and the upper index its reference system, i.e. the relatum.

Due to the ambiguousness of the given texts, each parameter belongs to a real valued interval. Usually there is no information on the exact extents of the objects. So we have to deal with intervals, e.g. the radius of the cat may range[3] in the interval $[0.1, 0.5]$.

The other spatial relations used, like `left`, `front`, and `behind` are defined in an analogous manner. Note that for the relation `right`, like for every spatial relation, the formulas differ depending on the shape of the relata and referents.

Since we use an intrinsic interpretation of the spatial relations, the constraints of each relation are defined wrt. a local reference system, i.e. the coordinate system of the relatum. That implies that we need to transform the coordinates of an object using the corresponding matrix when changing its relatum. A point $\mathbf{s}_k$, expressed wrt. system j, is transformed to the new reference system i using the homogeneous transformation matrix $\mathbf{P}_j^{\ i}$:

$$\mathbf{s}_k^i = \mathbf{P}_j^{\ i} \mathbf{s}_k^j$$

where

$$\mathbf{P}_j^{\ i} = \begin{pmatrix} \cos\Delta\theta_j^{\ i} & -\sin\Delta\theta_j^{\ i} & \Delta x_j^{\ i} \\ \sin\Delta\theta_j^{\ i} & \cos\Delta\theta_j^{\ i} & \Delta y_j^{\ i} \\ 0 & 0 & 1 \end{pmatrix}.$$

The parameter $\Delta\theta_j^{\ i}$ describes the rotation angle of system j wrt. system i. The distances of object j in the x- and y-directions from the origin of the relatum i are denoted by $\Delta x_j^{\ i}$ and $\Delta y_j^{\ i}$, resp.

3 Drawing Inferences

Given the system of, in general, nonlinear constraints and transformation matrices (see below) we can draw inferences, i.e. answer questions on relations between objects which where not initially given in the text. As shown in [Wiebrock *et al.*, 2000], this is done by propagating the constraints (i.e. (1) to (3) when considering `right(table, spots)`). The homogeneous transformation matrices are used for changing the reference system.

Suppose we want to verify, whether the relation `right(steffi, spots)` is consistent with the text given in Sect. 1 (Fig. 1). Since there are already constraints for the cat (expressed wrt. the system of the table), we transform them into the coordinate system of steffi using

[2]The theory and the methods described in this paper are applicable for 3D problems. For better understanding we use without restrictions 2D space only.

[3]This is some kind of background knowledge for the considered domain.

the transformation matrices $\mathbf{P}^{\text{steffi}}_{\text{table}}$ and $\mathbf{P}^{\text{table}}_{\text{spots}}$. Thus we have to calculate

$$\mathbf{P}^{\text{steffi}}_{\text{spots}} = \mathbf{P}^{\text{steffi}}_{\text{table}} \mathbf{P}^{\text{table}}_{\text{spots}}$$

and to compare the resulting matrix $\mathbf{P}^{\text{steffi}}_{\text{spots}}$ and its constraints with the defining constraints for `right`. For instance we have to check, whether

$$\begin{aligned} \Delta x^{\text{steffi}}_{\text{spots}} &= \cos\left(\Delta\theta^{\text{steffi}}_{\text{table}}\right) \Delta x^{\text{table}}_{\text{spots}} - \\ &\quad \sin\left(\Delta\theta^{\text{steffi}}_{\text{table}}\right) \Delta y^{\text{table}}_{\text{spots}} + \\ &\quad \Delta x^{\text{steffi}}_{\text{table}} \end{aligned}$$

meets the definition of `right` (cf. Fig. 3). Only in very rare cases ([Wiebrock *et al.*, 2000]) we can do the comparison analytically. Thus we have to use numerical methods to solve and check the constraints. In the following sections we apply methods of Machine Learning to this problem.

3.1 Learning the Spatial Relations

Each relation r_i corresponds to a region $\mathbf{G}_i$ in the configuration space[4], where all its constraints are satisfied. We try to learn a decision function which can decide whether a given vector of the configuration space belongs to the region of the relation. Afterwards we use the result from the learning step to solve the constraints and generate depictions. We introduced this idea in [Gips *et al.*, 2002] as an alternative approach for solving spatial constraints.

For each requested combination of relations and tuples of object types we construct training sets by exploiting the given constraints for the particular relation. Using these training data sets, Machine Learning algorithms construct classifiers. We have chosen the decision tree learning algorithm CAL5 ([Müller and Wysotzki, 1997]) for learning the spatial relations. It uses axis-parallel hyperplanes for approximating the class boundaries piecewise linearly. The main advantage of using decision trees is their interpretability. The paths in the decision tree contain the approximation of the solution region $\mathbf{G}_i$ of a relation r_i. This can be used in a constructive manner for deciding the constraints in later steps.

For a more detailed survey on the learning process and its results refer to [Gips *et al.*, 2002; Wiebrock, 2000]. Note, the learned decision trees contain all generalisations for a given tuple of object types. So we can use the same tree for different input objects of a relation, as long as the objects have the specific shapes the tree was trained for.

3.2 Inference using Sets

Using the learned decision trees we can draw inferences on relations between objects which were not initially given in the text. Given the relations `right(steffi, table)` and `right(table, spots)`, is there any other relation `relation` between Steffi and Spots? More generally, given the spatial relations $\mathrm{r}_1(O_1, O_2)$ and $\mathrm{r}_2(O_2, O_3)$, can we infer a third relation $\mathrm{r}_3(O_1, O_3)$? This can be expressed by

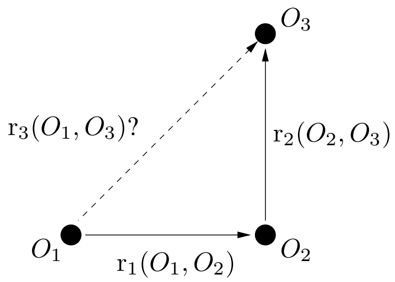
$$\forall O_1, \forall O_2, \forall O_3 : \; \mathrm{r}_1(O_1, O_2) \land \mathrm{r}_2(O_2, O_3) \rightarrow \mathrm{r}_3(O_1, O_3) \quad (4)$$

As mentioned above, it is seldom possible to prove the validity of those statements analytically. However, we can reduce the problem to intersecting sets in the configuration space.

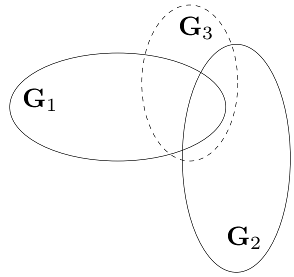
[4]The configuration space is the space where each parameter or variable of the problem spans a dimension.

The relations r_i (whereby each r_i is defined by a set of constraints) form a constraint net (Fig. 4(a)). Each relation can be interpreted as an area $\mathbf{G}_i$ in the configuration space. Note, that these areas $\mathbf{G}_i$ are given in the local reference system of the relation r_i, i.e. the coordinate system of the first argument object of r_i. However, for intersecting the areas $\mathbf{G}_i$ we have to transform them into the same, i.e. a global coordinate system (e.g. the coordinate system associated with O_1 or the room system). In order to infer r_3 in (4) from r_1 and r_2, the intersection of $\mathbf{G}_1$ and $\mathbf{G}_2$ would have to lie completely within $\mathbf{G}_3$ (Fig. 4(b)):

$$\mathbf{G}_1 \cap \mathbf{G}_2 \subseteq \mathbf{G}_3.$$



(a) Constraint net



(b) Intersection of the solution regions (ideal case)

Figure 4: Inference in a set context

If the intersection $\mathbf{G}_1 \cap \mathbf{G}_2$ is not completely contained in $\mathbf{G}_3$, certain constraints $\mathrm{c}(O_1, O_3)$ have to be added to the premise to state, under which conditions the conclusio can be inferred, reducing the $\mathbf{G}_i$:

$$\begin{aligned} \forall O_1, \forall O_2, \forall O_3 : \; &\mathrm{r}_1(O_1, O_2) \land \mathrm{r}_2(O_2, O_3) \land \\ &\mathrm{c}(O_1, O_3) \rightarrow \mathrm{r}_3(O_1, O_3) \end{aligned} \quad (5)$$

The new constraint $\mathrm{c}(O_1, O_3)$ denotes a conjunction of constraints for O_1 and O_3. Note, that almost every conclusio depends on the relative position and orientation of the coordinate systems. Thus we will usually get the non-ideal case (5) and have to add the additional constraints $\mathrm{c}(O_1, O_3)$ to the premise. These constraints are a result of our new algorithm (cf. Sect. 3.4).

Using the learned CAL5-decision trees we can avoid the complex analytical calculation of the constraints. Instead we can intersect the solution regions of the relations, which can be extracted from the decision trees. This is done by exploration of the intersection using an algorithm presented in [Gips *et al.*, 2002] and the learned trees. In Sect. 3.4 we show, how to prove the implication and how to quantify the additional constraints, if needed, using the exploration and, again, CAL5.

3.3 Generating Datasets for Satisfying the Premise

As mentioned above, inference of a relation $\mathrm{r}_3(O_1, O_3)$ from a proven knowledge $\{\mathrm{r}_1(O_1, O_2), \mathrm{r}_2(O_2, O_3)\}$ means to prove the implication (4).

The first task is to find (theoretically all) objects O_1, O_2 and O_3, so that the premise $\mathrm{r}_1(O_1, O_2) \wedge \mathrm{r}_2(O_2, O_3)$ is valid. We have to find values for the x- and y-parameters, the rotation angle of the objects and values for their extents (i.e. width and depth or radius) so that all the constraints of the relations r_1 and r_2 are satisfied. Such a combination of valid parameters corresponds to a depiction of the conjunction of the relations r_1 and r_2.

In [Gips *et al.*, 2002] we presented a heuristic backtracking search algorithm, which used the learned CAL5-decision trees ([Müller and Wysotzki, 1997]) for finding an appropriate depiction for a given set of spatial relations. This algorithm is used now for generating a number of scenes, i.e. sets of objects O_1, O_2 and O_3, which satisfy the premise $\mathrm{r}_1(O_1, O_2) \wedge \mathrm{r}_2(O_2, O_3)$. Theoretically we would have to generate an infinite number of such depictions for actually finding all objects, which satisfy the premise, i.e. to explore the complete intersection of the solution regions of the relations in the premise. For practical reasons we actually generate only about 1.000 such scenes.

3.4 Conclusio: Proving the Implication

Given the data set satisfying the premise (generated by the depiction algorithm described in the previous section), we have to check $\mathbf{G}_1 \cap \mathbf{G}_2 \subseteq \mathbf{G}_3$. The data set consists of vectors, which represent objects from the premise with their position, orientation and also their size, i.e. a sample of $\mathbf{G}_1 \cap \mathbf{G}_2$. Thus each vector corresponds to an appropriate depiction of the given spatial description and satisfies the premise, i.e. each vector belongs to $\mathbf{G}_1 \cap \mathbf{G}_2$.

For classical inference all the vectors should be completely contained in the solution region of the relation to be inferred. This can be proved quickly by classifying all vectors using the learned decision tree for the relation to be inferred (r_3). If all vectors are classified correctly, i.e. they satisfy r_3 and, thus, lie in $\mathbf{G}_3$, we can draw the inference. This is the ideal case: we do not need any additional constraints.

Otherwise we must define (in contrast to the classical logic) some new constraints, so that the inference is valid. Therefore we use the generated set of depictions as a training set: we classify each vector whether it satisfies the relation to be inferred (class 'A') or not (class 'B'). Using this training set and CAL5 we generate a new decision tree. From this tree the additional constraints $\mathrm{c}(O_1, O_3)$ for drawing the inference $\mathrm{r}_3(O_1, O_3)$ can be directly obtained simply by reading off the paths to end nodes marked with class 'A'.

In the following section we apply this algorithm for proving the implication and for finding possibly needed additional constraints to some example problems.

4 Experiments

As mentioned in the previous section, the first step is to generate a dataset satisfying the premise. Therefore we employ the depiction generating algorithm presented in [Gips *et al.*, 2002]. It uses the learned decision trees for the spatial relations to find points in the configuration space, which satisfy the relations in the premise. These points build up the dataset. In the next step the dataset is classified using the decision tree of the relation to be inferred. If all points in the dataset satisfy the conclusio, we can state, that this relation can be inferred without any additional constraints. In the case there isn't any point in the dataset, which satisfy the conclusio, the relation cannot be inferred. Otherwise we start CAL5 on the dataset and train a new decision tree. From this tree we can extract additional constraints for a possible inference of the conclusio.

A first decision is to choose the optimal size of the dataset, i.e. the number of depictions used for the experiments. This is a compromise between accuracy, i.e. training and generalisation error, and calculation time, i.e. time to learn the decision tree. For the experiments presented in this paper a dataset size of 1.000 scenes proved as a good compromise.

The constraints, and thus the learned trees, depend on the definition of the spatial relations. As shown in Sect. 2 we usually use sectors defined by bisecting lines of certain quadrants (i.e. the first and second quadrant for `right`, cf. Fig. 3). For the first presented experiment (Sect. 4.1) we have chosen this variant. In order to compare the results and the predictions for a more complex example, we used a simpler definition of the relations, called 'strict relations', in the second experiment (Sect. 4.2). In the case of the `right` relation this means that the sector defined by the constraints (1) to (3) is replaced by a single line. The centre of the second object must be somewhere on the positive x-axis of the first object, i.e. $\Delta x_b{}^a \geq a.w + b.r$ and $\Delta y_b{}^a = 0$.

4.1 Two Objects, One Relation

To pick up a simplification of the initial example from Sect. 1 we assume the validity of the relation `left(steffi, table)`, i.e. there is a single relation in the premise. From this premise we can derive successively the relations `right`, `behind`, `left`, and `front` for `(table, steffi)`. For each conclusion we obviously have to restrict the angle between `steffi` and `table`, i.e. $\Delta\theta_{\text{steffi}}^{\text{table}}$, as additional constraint. Note, the relations `right(steffi, table)`, `front(steffi, table)`, and `behind(steffi, table)` cannot be concluded.

In order to conclude the relation `right(table, steffi)` from the premise `left(steffi, table)` the objects `steffi` and `table` need to have a similar orientation. Since we use relations defined like `right` in Sect. 2, the inference algorithm should theoretically produce the additional constraint c_1`(steffi, table)` = $\Delta\theta_{\text{steffi}}^{\text{table}} \in [-\pi/4, +\pi/4]$ resp. approximate it with intervals in certain paths of a decision tree. I.e. the relative orientation $\Delta\theta_{\text{steffi}}^{\text{table}}$ does not have to be zero but spans an interval of $\pm 1/4\pi$ around zero. This is equivalent to $\Delta\theta_{\text{steffi}}^{\text{table}} \in [5.50, 6.28]$ and $\Delta\theta_{\text{steffi}}^{\text{table}} \in [0.00, 0.79]$, because $\Delta\theta_{\text{steffi}}^{\text{table}}$ is an angle. The universally valid inference of `right(steffi, table)` is thus:

$$\forall\,\text{steffi}, \forall\,\text{table} : \text{left}(\text{steffi}, \text{table}) \wedge$$
$$\Delta\theta_{\text{steffi}}^{\text{table}} \in [-\frac{1}{4}\pi, +\frac{1}{4}\pi] \quad \rightarrow \quad \text{right}(\text{table}, \text{steffi})$$

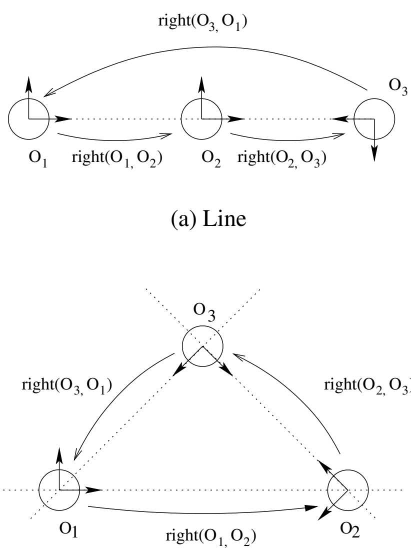
Table 1 shows the mean values of the interval borders of the inference results of 30 passes in comparison to the theoretical expectations. Each pass is based on 1.000 scenes.

Table 1: Prediction for $\Delta\theta_{\text{steffi}}^{\text{table}}$ and the experimental results in 30 runs

Relation	Prediction $\Delta\theta_{\text{steffi}}^{\text{table}} \in$	Result $\Delta\theta_{\text{steffi}}^{\text{table}} \in$
`right(table, steffi)`	[5.50, 6.28]	[5.19, 6.26]
	[0.00, 0.79]	[0.02, 0.83]
`behind(table, steffi)`	[0.79, 2.36]	[1.17, 2.27]
`left(table, steffi)`	[2.36, 3.93]	[2.72, 4.14]
`front(table, steffi)`	[3.93, 5.50]	[4.21, 5.45]

4.2 Three Objects, Three Strict Relations

The complexity of the inference is now extended to form a ring of three circle objects O_1, O_2 and O_3. The premise chains these objects with the three relations `right(`O_1`,` O_2`)`, `right(`O_2`,` O_3`)` and `right(`O_3`,` O_1`)`[5]. From this configuration the four spatial relations (`right`, `behind`, `left`, `front`) for each remaining combination of objects $\{(O_1, O_3), (O_3, O_2), (O_2, O_1)\}$ can be concluded. Again, each dataset contained 1.000 vectors, i.e. scenes, and for each inferred relation 10 runs were executed with strict relations. For easier predictions we use strict relations in this example.



(a) Line

(b) Triangle

Figure 5: Alignments of rings with three objects chained with `strictly right` relations

When generating scenes using the strict relations, the solution area for each relation forms an ϵ-stripe around one of the objects intrinsic axis. Only two general cases of object alignment occur when inference is drawn (Fig. 5) and each inferred relation falls into one of these categories and has therefore a predictable $\Delta\theta$ (Tab. 2). If the objects are aligned in a line for example (Fig. 5(a)), the second object O_2 is `strictly right` of the first and the third object O_3 is `strictly right` of the second. O_3 has to be rotated by exactly π relative to O_1 so that O_1 enters the `right` area of O_3!

[5] We use this relations for getting a closed loop, which enforces a small set of possible configurations and, thus, enables predictions for inferences.

4.3 "Knights of the Round Table" Scene

To test the functionality with more complex scenes a "Knights of the Round Table" configuration was generated in which six knights were placed around a table. The following relations form the premise:

$$\texttt{right}(K_1, K_2) \wedge \texttt{right}(K_2, K_3) \wedge$$
$$\texttt{right}(K_3, K_4) \wedge \texttt{right}(K_4, K_5) \wedge$$
$$\texttt{right}(K_5, K_6) \wedge \texttt{right}(K_6, K_1) \wedge$$
$$\texttt{front}(K_n, \texttt{ table});\, n = 1, \ldots, 6$$

Figure 6: Premise for a "Knights of the Round Table" scene

The K_n represent the six knights, randomly generated with a radius $r \in [0.2, 0.3]$, the table is defined as a circle with $r \in [0.3, 0.5]$. As in the first experiment, the area based relations are used. Due to the complexity of this configuration (6 objects, 12 relations) the inference is based on 200 scenes only.

The knights form a circle around the table, because they have to face the table and each knights successor is placed to his right. Thus the predecessor of a knight will often be positioned to his left. For inference we have to ensure, that the concluded relations `left(`K_1, K_6`)`, `left(`K_6, K_5`)`, ..., `left(`K_2, K_1`)` are always valid. For this our inference algorithm found reasonable constraints.

As the opposite knights both look at the table and so subsequently look at each other, the `front` relation for them should be inferred as always true and `behind` as always false. These conclusions are universally valid and no constraints may be generated. This assumption was validated by the algorithm.

5 Discussion

In this paper we presented an approach for spatial inference, where spatial relations are represented by parameterised transformation matrices and constraints on the parameters. Inference of relations not mentioned in the spatial description is done by propagating the constraints. Thereby we have to change the reference systems because of the intrinsic definition of the spatial relations and, thus, the local definition of the constraints. We use therefore homogeneous transformation matrices, which leads to a system of nonlinear constraints.

Instead of solving this constraint net directly, we employed methods of Machine Learning. We used the decision tree learner CAL5 to transform the constraint solving problem into a classification problem. Using CAL5 yields well interpretable results. After learning we have, due to the obtained decision trees, detailed knowledge about the regions in the configuration space. Inference of a spatial relation from a premise, i.e. a set of relations, is now done by intersecting the solution regions of the relations in the

Table 2: Prediction of $\Delta\theta$, experimental results in 10 runs

Relation	Prediction $\Delta\theta$	Result $\Delta\theta$
left(O_1, O_3)	$\Delta\theta_{O_3}^{O_1} = 0 = 6.28$	$\Delta\theta_{O_3}^{O_1} \in []$
	$\Delta\theta_{O_3}^{O_1} = 2\pi = 6.28$	$\Delta\theta_{O_3}^{O_1} \in [6.04, 6.27]$
right(O_1, O_3)	$\Delta\theta_{O_3}^{O_1} = \pi = 3.14$	$\Delta\theta_{O_3}^{O_1} \in [3.13, 3.25]$
front(O_1, O_3)	$\Delta\theta_{O_3}^{O_1} = 3/2\pi = 4.71$	$\Delta\theta_{O_3}^{O_1} \in [4.62, 4.94]$
behind(O_1, O_3)	$\Delta\theta_{O_3}^{O_1} = 1/2\pi = 1.57$	$\Delta\theta_{O_3}^{O_1} \in [1.42, 1.74]$
left(O_3, O_2)	$\Delta\theta_{O_2}^{O_3} = 0 = 6.28$	$\Delta\theta_{O_2}^{O_3} \in []$
	$\Delta\theta_{O_2}^{O_3} = 2\pi = 6.28$	$\Delta\theta_{O_2}^{O_3} \in [6.13, 6.26]$
right(O_3, O_2)	$\Delta\theta_{O_2}^{O_3} = \pi = 3.14$	$\Delta\theta_{O_2}^{O_3} \in [3.13, 3.30]$
front(O_3, O_2)	$\Delta\theta_{O_2}^{O_3} = 3/2\pi = 4.71$	$\Delta\theta_{O_2}^{O_3} \in [4.71, 4.99]$
behind(O_3, O_2)	$\Delta\theta_{O_2}^{O_3} = 1/2\pi = 1.57$	$\Delta\theta_{O_2}^{O_3} \in [1.46, 1.71]$
left(O_2, O_1)	$\Delta\theta_{O_1}^{O_2} = 0 = 6.28$	$\Delta\theta_{O_1}^{O_2} \in [0.01, 0.21]$
	$\Delta\theta_{O_1}^{O_2} = 2\pi = 6.28$	$\Delta\theta_{O_1}^{O_2} \in [6.12, 6.27]$
right(O_2, O_1)	$\Delta\theta_{O_1}^{O_2} = \pi = 3.14$	$\Delta\theta_{O_1}^{O_2} \in [3.13, 3.28]$
front(O_2, O_1)	$\Delta\theta_{O_1}^{O_2} = 3/2\pi = 4.71$	$\Delta\theta_{O_1}^{O_2} \in [4.72, 4.91]$
behind(O_2, O_1)	$\Delta\theta_{O_1}^{O_2} = 1/2\pi = 1.57$	$\Delta\theta_{O_1}^{O_2} \in [1.52, 1.74]$

premise. In the ideal case the solution region of the inferred relation should be completely contained in this intersection. Otherwise we get additional constraints for inferring this relation from the given premise.

For calculation of the intersection of the premise relations we use an heuristic search algorithm ([Gips *et al.*, 2002]). Using the learned CAL5-decision trees it computes a set of depictions for a given set of spatial relations, i.e. the premise. These depictions correspond to a set of appropriate objects which satisfy all the relations of the premise. Thus the vector of the object parameters belongs to the intersection region of the premise relations. For exploring the complete intersection we would have to generate an infinite number of such depictions. For practical reasons we actually use only about 1.000 scenes.

This set of depictions, i.e. the set of object vectors satisfying the premise, should be completely contained in the solution region of the relation to be inferred. This can be proved quickly by classifying all vectors using the learned decision tree for the inferred relation. If all vectors are classified correctly, we can draw the inference. Otherwise we can define (in contrast to the classical logic) some new constraints, so that the inference is valid. Therefore we use the generated set of depictions as training set: we classify each vector whether it satisfies the relation to be inferred. Using this training set and CAL5 we generate a new decision tree. From this tree the additional constraints for drawing the inference can be extracted.

In the experiments shown the results of our approach come very close to the theoretical (predicted) values. Even for the complex problem of the "Knights on the Round Table" we got reasonable results.

Future work includes a deeper investigation of this approach, including a speed up of the algorithm. Because of the trade-off between computing time and precision we have to find values for an optimal compromise for the actually needed number of generated depictions. Furthermore we have to compare our approach to classical constraint solvers, like the RealPaver interval solver ([Granvilliers, 2003]).

References

[Cohn and Hazarika, 2001] A. Cohn and S. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46:2–32, 2001.

[Gips *et al.*, 2002] C. Gips, P. Hofstedt, and F. Wysotzki. Spatial Inference – Learning vs. Constraint Solving. In M. Jarke, J. Köhler, and G. Lakemeyer, editors, *KI 2002: Advances in Artificial Intelligence: 25th Annual German Conference on AI, KI 2002, Aachen, Germany, September 16-20, 2002*, volume 2479 of *LNAI*, pages 299–313. Springer, 2002.

[Granvilliers, 2003] L. Granvilliers. Realpaver, 07 2003. Available from `www.sciences.univ-nantes.fr/info/perso/permanents/granvil/realpaver/`.

[Müller and Wysotzki, 1997] W. Müller and F. Wysotzki. The Decision-Tree Algorithm CAL5 Based on a Statistical Approach to Its Splitting Algorithm. In G. Nakhaeizadeh and C. C. Taylor, editors, *Machine Learning and Statistics - The Interface*, pages 45–65. Wiley, 1997.

[Wiebrock *et al.*, 2000] S. Wiebrock, L. Wittenburg, U. Schmid, and F. Wysotzki. Inference and Visualization of Spatial Relations. In C. Freksa, W. Brauer, C. Habel, and K. Wender, editors, *Spatial Cognition II*, LNAI 1849. Springer, 2000.

[Wiebrock, 2000] S. Wiebrock. Anwendung des Lernalgorithmus CAL5 zur Generierung von Depiktionen und zur Inferenz von räumlichen Relationen. Technical Report 2000-14, ISSN 1436-9915, TU Berlin, 2000.