SPATIAL INFERENCE AND CONSTRAINT SOLVING

How to Depict Textual Spatial Descriptions from Internet

Carsten Gips
Berlin University of Technology
cagi@cs.tu-berlin.de

Fritz Wysotzki
Berlin University of Technology
wysotzki@cs.tu-berlin.de

Abstract Today there are still many applications in the Internet, where the user is given a textual description of a spatial configuration (e.g. chat, e-mail or newsgroups). The user is asked to imagine the scene and to draw inferences. We present a new approach to generate depictions of such scenes. Besides of drawing spatial inferences, this leads to the problem of solving a system of complicated numerical constraints. In contrast to qualitative spatial reasoning, we use a metric description where relations between pairs of objects are represented by parameterized homogenous transformation matrices with numerical (nonlinear) constraints. We employ methods of machine learning in combination with a new algorithm for generating depictions from text including spatial inference.

Keywords: Spatial Reasoning, Constraint Satisfaction, Machine Learning, Depictions

1. Introduction

There are many fields where it is important to understand and interpret textual descriptions of real world scenes. Examples are navigation and route descriptions in robotics (Röfer, 1997; Jörding and Wachsmuth, 1996), CAD and graphical user interfaces (e.g. "The xterm is right of the emacs.") or visualization of scenes given in the Internet (e.g. in newsgroups or in e-mail).

In contrast to qualitative approaches to spatial reasoning (Allen, 1983; Guesgen, 1989; Hernández, 1994), we presented in Claus et al., 1998 a new metric approach to spatial inference based on mental models (Johnson-Laird, 1983). Starting from textual descriptions containing sentences like "The lamp is left of the fridge." we try to construct a mental model which represents the described spatial situation. This approach uses a directed graph, where the nodes represent the objects and the edges represent the given relation between two objects, e.g. $\mathrm{left}(\mathrm{fridge}, \mathrm{lamp})$. From this model it is possible to infer relations which were not initially given in the text (i.e. to answer questions about the described spatial scene or to complete the model). In a further step we can use the model to generate depictions compatible with the description.

The semantics of the relations is given by homogenous transformation matrices with constraints on the variables. As shown in Wiebrock et al., 2000, inference of a relation between two objects is done by searching a path between the objects and multiplying the matrices on this path. Thereby constraints containing inequalities and trigonometric functions must be propagated and verified. Only in some rare cases we can solve these constraints analytically. Wiebrock et al., 2000 proposed a simple algorithm for generating depictions. It is restricted to default positions of objects and to rotations of multiples of $\pi/2$. Moreover, they had to keep lists with possible positions for every object.

Our aim is to find a method to solve this kind of constraints and to generate depictions without these restrictions. We sketch an approach to spatial reasoning which applies machine learning in combination with a new algorithm for depiction generation.

This paper is structured as follows: We start with an introduction into the description of spatial relations in Sect. 2. In Sect. 3 we apply methods of machine learning to learn the semantics of the relations, in order to obtain an alternative representation of the constraints. Afterwards we sketch in Sect. 4 a new approach for generating depictions of the described spatial layout, i.e. solving the constraints, which uses the results of the machine learning step. At the end of the article we give in Sect. 5 a conclusion and draw some research perspectives.

2. Expressing Spatial Relations

Starting from texts with descriptions of spatial layouts, we want to generate appropriate depictions. Additionally we have to determine whether the given descriptions are inconsistent, i.e. whether there are no possible depictions.

The texts describe the scenes by the use of spatial relations. We investigated scene descriptions based on the relations $\mathrm{left}/2$ and $\mathrm{right}/2$, which describe the placement of an object left resp. right of another one, the relations $\mathrm{front}/2$ and $\mathrm{behind}/2$ which place objects in front of or behind other objects, resp., and the relation $\mathrm{atwall}/2$ for describing the placement of an object parallel to a wall with a fixed maximum distance. Further relations provide background knowledge (i.e. an object is always situated in a given room and the objects must not overlap). For simplification, we consider 2D scenes only and represent objects by appropriate geometric figures.

As mentioned above, in contrast to qualitative techniques (Allen, 1983; Guesgen, 1989; Hernández, 1994) we use a metric approach for spatial reasoning (Claus et al., 1998; Geibel et al., 1998) known from the area of robotics (Ambler and Popplestone, 1975). We associate with every object a coordinate system, and its form and size. Relations between pairs of objects are represented by constraints on parameters of their transformation matrices. Thus, the current coordinates of an object are expressed relative to the coordinate system of its relatum, which may be different in different relations. That means, when changing the relatum of an object, we need to transform the coordinates of this object by multiplying them with the corresponding matrix.

Let us consider the relation $\mathrm{right}/2$ in detail. Called with cupboard and lamp as arguments ($\mathrm{right}(\mathrm{cupboard}, \mathrm{lamp})$), it places the lamp, which is the referent, right wrt. its relatum, the cupboard. Thus the coordinate system of the cupboard is the origin of the relation. The lamp can be placed in the area restricted by the upper and lower bisectors of the upper and lower right corner and the right side of the rectangle. Figure 1 illustrates this situation.

Mathematically we can describe the relation $\mathrm{right}(O1, O2)$[1] by the inequalities 1a, 1b, and 1c.

$$
\begin{aligned}
\Delta {x_2}^1 &\geq O1.w + O2.r && \text{(1a)}\\
\Delta {x_2}^1 &\geq \Delta {y_2}^1 + O1.w - O1.d + \sqrt{2}\, O2.r && \text{(1b)}\\
\Delta {x_2}^1 &\geq -\Delta {y_2}^1 + O1.w - O1.d + \sqrt{2}\, O2.r && \text{(1c)}
\end{aligned}
$$

Thereby $O1.w$ and $O1.d$ represent the width and the depth of the rectangle, i.e. the cupboard, and $O2.r$ stands for the radius of the lamp. The distances of the object $O2$ in the x- and y-directions from the relatum $O1$ are denoted by $\Delta {x_2}^1$ and $\Delta {y_2}^1$, resp. Each object has intervals for its variables, because it stands for a class of objects (i.e. the object $\mathrm{cupboard}$ is a "cupboard frame", standing not only for a particular cupboard but for all possible cupboards).

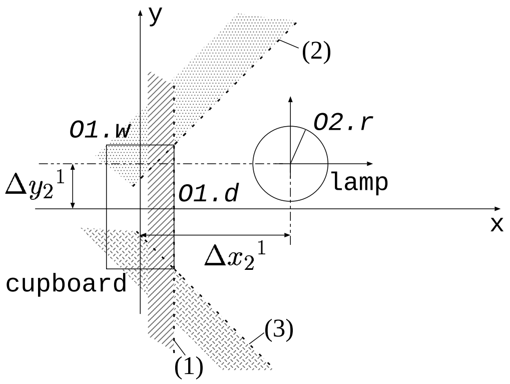
Note, that for the relation $\mathrm{right}/2$, like for every spatial relation, the formulae differ generally depending on the form of the relata and referents.

3. Learning Spatial Relations with CAL5

In our problem domain (objects to be located in a room), the constraints consist of equations and inequalities containing trigonometric functions which lead to computational difficulties well known from robotics (Ambler and Popplestone, 1975).

Instead of solving the constraints directly we try to learn the decision function $C(x_1, \ldots, x_n)$ which decides whether a vector $x = (x_1, \ldots, x_n)$ of the

[1] $O1$ and $O2$ may stand for the cupboard and the lamp, resp.



PSfrag replacements

Figure 1. The relation $\mathrm{right}(\mathrm{cupboard}, \mathrm{lamp})$ in detail

configuration space belongs to a region where the predicate C is true, i.e. the corresponding constraints are satisfied.

Before employing machine learning algorithms we have to construct a training set by exploiting the given constraint description or by using results of psychological studies. These datasets consist of preclassified feature vectors where each variable of the constraints represents an attribute, i.e. a dimension in the feature space. In the following, we will use “class A” for the regions where a constraint C is satisfied (and “class B” otherwise). By means of the training sets, algorithms of classification learning (e.g. decision tree learning like CAL5 or neuronal nets like Dipol, see Müller and Wysotzki, 1997 or Schulmeister and Wysotzki, 1994, resp.) construct classifiers. These decide the class membership of an arbitrarily chosen point x (not necessarily contained in the training set) by inductive generalization. Generating training sets in order to get an acceptable approximation of the decision boundary is also known as “learning by exploration” or “active learning” in literature.

In this work we have chosen CAL5 for learning the spatial relations. CAL5 approximates, as it is a decision tree learner, the class boundaries piecewise linearly by axis-parallel hyperplanes. Usually, there is a generalization error due to the unavoidable approximation of the boundaries between the A-regions and the B-regions. This error can be measured using a test set of classified example vectors different from the training set. By increasing the number of training data (and by simultaneously shrinking a certain parameter of CAL5) the generalization error can be reduced (i.e. the accuracy of the class boundary approximation can be made arbitrarily high), and in the limit of an infinite set of training data the error becomes zero. In Geibel et al., 1998 we investigated the problem “a bar is right of an object O”, represented in the configuration space defined by the angle of the bar with the x-axis and the displacement of the bar with respect to the origin S of the coordinate system of the relatum O. We also demonstrated experimentally, that the generalization error of the obtained decision tree shrinks with an increasing number of points for learning. However, in practice we reach the manageable limit at 200.000 training examples. The constraints of our relations (like $\mathrm{right}/2$ for $\mathrm{circles}$ and $\mathrm{rectangles}$, see Sect. 2) affect up to seven parameters, thus, we obtain a configuration space with up to seven dimensions. This corresponds to approximatively ten data points per dimension[2]. Because of this sparsely populated configuration space both the training and generalization errors are rather high. This is shown in Tab. 1, where we used 200.000 uniformly distributed data points for learning and 5.000 points for testing.

Relation[3]	*Number of class A leafs*	*Points in A*	*Points in B*	*Test error for A only*	*Test error for B only*	*Test error total*
$\mathrm{atwall}(r, r)$	143	15.909	184.091	10 %	1 %	2 %
$\mathrm{front}(c, r)$	1.984	84.285	115.715	22 %	14 %	17 %
$\mathrm{right}(c, r)$	1.702	84.557	115.443	20 %	13 %	16 %
$\mathrm{right}(r, c)$	2.079	87.550	112.450	26 %	14 %	19 %

Table 1. Results of the learning process for some spatial relations

Furthermore we have to be aware of generating points in a sufficient large subspace of configuration space. Figure 2 shows the result of a too small scope of the training data. The resulting classifier does not cover the intersection of the $\mathrm{right}/2$ sector with the room area.

The benefits of our learning approach are to get a new, easier representation of the decision boundary (i.e. the constraints). The new representation contains the solution of the constraints (i.e. the A-regions), and the accuracy of the approximation can be made arbitrarily high. The problem, however, is the generation of suitable data sets.

4. Generating Depictions

In the previous section we transformed the constraints of the spatial relations into a new representation (i.e. the learned classifiers). These CAL5 decision

[2] Consider the relation $\mathrm{right}/2$ for two rectangles. This problem has seven parameters, so the configuration space consists of seven dimensions. Supposed we obtain the same number of data on each dimension (like a grid), the 7th root of 200.000 yields approximately six data points on each dimension. Note, this is not a correct calculation but a simple estimation.

[3] r denotes a $\mathrm{rectangle}$, and c a circle, resp.

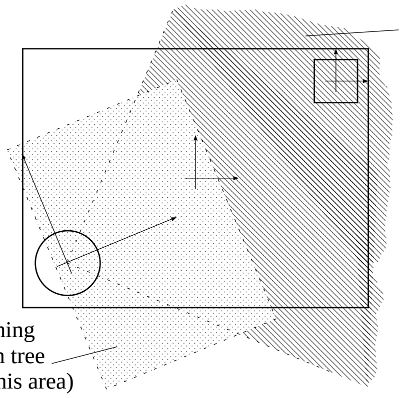


Figure 2. Scope of training data not adapted to the admeasurement of the room, the classifier may not cover the intersection of the $\mathtt{right}/2$ sector with the room area

trees are used by a new algorithm (see Fig. 3) for generating depictions, i.e. for simultaneously solving the set of given spatial constraints.

As mentioned above, for every needed relation and for every pair of object types, corresponding (sub-)relations must be learned. Recall that our relations are binary. Thus, we get three cases: both objects are ‘unknown’, one object is already placed or both objects are placed. In the first case, we place one object randomly in the room (line 11 in Fig. 3). This leads us to the second case. There we pick a class A leaf of the tree and compute the size, relative position and relative orientation of the other object by assigning values to the remaining variables of this object within the intervals of the chosen leaf (line 12 and 13). If the collision check in line 14 fails (for both, case one and two), we repeat the procedure up to k times. If we do not have admissible values after the kth trial, we suppose that the current relation cannot hold in combination with the others and reject the depiction generated so far. However, there may exist solutions. Actually we cannot distinguish between the case “no solution” and “disadvantageous values”. So if we reject the vector[4], we have to start again with the first relation. For practical reasons we work instead on a number of object constellations[5] in parallel. In the last case both objects are already placed and we have to check, wether the values of the objects range in the intervals represented by at least one leaf (line 05 to 07). If not, the relations do not hold, at least for the calculated values.

[4]Note, that each depiction is a point in the configuration space. This points are represented by vectors.
[5]Initially we have a number v of empty vectors.

Depiction generation algorithm

INPUT:

number v of initial vectors and number k of trials
relations r that have to hold

OUTPUT:

up to v depictions, where all relations r hold

ALGORITHM:

```
 foreach relation r:
   identify objects and object types by object descriptions
   load the corresponding decision tree
    foreach vector v:
       if both objects were placed
       then
          check whether relation r holds
       else
           if both objects are new
           then
              place first object randomly in room
          pick randomly class A leaf
          assign values to variables within intervals of leaf
          check non-overlapping with other objects and walls
           repeat up to k times if check fails
       if no success
       then
          drop vector v
show remaining vectors (depictions)
```

Figure 3. Algorithm for generation depictions using decision trees

This procedure is repeated for every relation with the remaining objects, which satisfy the relations processed in the former steps. Finally we obtain up to v depictions according to the given spatial description. In the case, that we have not found any depiction, we have to assume that the constraints are unsatisfiable.

Up to now each decision tree represents only the constraints for the particular relations with the two objects of the specified forms. The background knowledge[6] was not learned but will be checked after every step explicitly. In

[6] E.g. the objects must not overlap.

general, it is possible to learn the background knowledge constraints as well and to check them like the other relations.

The input parameters v and k depend on the number and on the type of the given relations. They have to be chosen large enough to get a correct answer ("There is no solution." or "We have found at least one.") with some probability. At the same time, one should choose rather small v and k, because by increasing the values of the parameters the calculation time increases, too. So they have to be chosen in relation to the problem to solve. As shown in Tab. 2, the more relations to solve, the higher v has to be.[7] A value of 100 for k seems to be a good choice. The number of trials per valid solution increases exponentially in the number of relations to be solved.

Not shown, but critical is the processing sequence. Restrictive relations like $\mathrm{atwall}/2$ should be solved at the beginning. Supposed we have two relations, $\mathrm{right}(\mathrm{cupboard}, \mathrm{lamp})$ and $\mathrm{atwall}(\mathrm{wall1}, \mathrm{cupboard})$. Now we fulfill first the $\mathrm{right}/2$ relation. Therefore the cupboard may be placed somewhere in a relatively large area in the room. After that we may be unable to satisfy $\mathrm{atwall}/2$, just because the cupboard should be placed nearby $\mathrm{wall1}$, but actually it is already placed at another area in the room. So we would have to increase v, but nevertheless the probability to get a solution is very small.

5. Discussion

In the previous sections we sketched a new approach to solve the constraints occuring in spatial reasoning. Instead of solving the constraints directly, we employed methods of machine learning for transforming the constraints into another representation. Using the decision tree learner CAL5 yields interpretable results. The approximation of the decision boundaries may be (at least in principle) arbitrarily high. Generating suitable training sets, however, is not trivial and is subject of current research ("active learning", Wiebrock and Wysotzki, 1999). So we have to deal with an increasing amount of disk space and calculation time and have to care for a suitable distribution of the training data.

After learning we have, due to the obtained decision trees, detailed knowledge about the regions in the configuration space. The depiction generation algorithm employs the decision rules for restricting the space to find possible solutions. In the limit of generating an infinite number of depictions (i.e. exhaustive search) the algorithm finds every possible solution. Because the processing sequence of the relations is critical we may find no solution, although there is one. However, the scene descriptions in this problem domain are usually underconstrained, and, thus, it is usually not a problem to find an alternative solution.

[7] For testing we implemented a first prototype in Perl, which is quite slow in comparison to usual programming languages, like C or Java. So we forgo to show the running times.

Combination of relations	*number v of initial vectors per valid depiction (average)*
single relation, e.g. $\mathrm{right(steffi, cupboard)}$	2
two relations, e.g. $\mathrm{right(steffi, cupboard)}$ $\mathrm{front(steffi, fridge)}$	10
three relations, e.g. $\mathrm{right(steffi, cupboard)}$ $\mathrm{left(fridge, lamp)}$ $\mathrm{right(cupboard, lamp)}$	61
four relations, e.g. $\mathrm{right(steffi, cupboard)}$ $\mathrm{left(fridge, lamp)}$ $\mathrm{right(cupboard, lamp)}$ $\mathrm{front(steffi, fridge)}$	375

Table 2. Some test runs and typical results of our algorithm

by constructing another sequence (i.e. following another path in the problem space).

This yields further research perspectives: First of all, we could use traditional constraint solving systems, like Hofstedt, 2000, for restricting the search space and for reducing the training cost. They could be employed for precomputing to get a sure exclusion of unsatisfiable scene descriptions (Gips et al., 2002). However, because of the incompleteness of the solvers, they will not detect all inconsistencies. Instead they yield regions, which are too large, but include the searched solution areas. So we could use the results of the constraint solvers for generating better training sets with fewer data points. Secondly, we have to investigate strategies for selecting a particular leaf of a decision tree in the depiction algorithm. Until now we use the volume of the region described by a leaf for selecting a particular leaf for further calculation. Recent studies (Wiebrock, 2000) have shown that a better selection criterion could dramatically improve the speed. Furthermore, we have to deal with the processing sequence of the relations. As explained, very restrictive relations should be satisfied first. On the basis of the decision trees we have to develop a measure for the restrictiveness. Thirdly, we should consider the principle of the least astonishment. We visualize spatial descriptions and draw inferences on the behalf of the user. So we should return those depiction (if there is any), which the user is most accustomed to. This could be achieved by using specific distributions for the training data, as first results in Wiebrock, 2000 show.

References

Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26:832–843.

Ambler, A. P. and Popplestone, R. J. (1975). Inferring the Positions of Bodies from Specified Spatial Relationships. *Artificial Intelligence*, 6:157–174.

Claus, B., Eyferth, K., Gips, C., Hörnig, R., Schmid, U., Wiebrock, S., and Wysotzki, F. (1998). Reference Frames for Spatial Inferences in Text Comprehension. In Freksa, C., Habel, C., and Wender, K. F., editors, *Spatial Cognition*, LNAI 1404. Springer.

Geibel, P., Gips, C., Wiebrock, S., and Wysotzki, F. (1998). Learning Spatial Relations with CAL5 and TRITOP. Technical Report 98-7, TU Berlin.

Gips, C., Hofstedt, P., and Wysotzki, F. (2002). Spatial Inference – Learning vs. Constraint Solving. In Jarke, M., Köhler, J., and Lakemeyer, G., editors, *Proceedings of the 25. German Conference on Artificial Intelligence, Sept. 16-20, 2002, Aachen, Germany*, LNAI. Springer. to appear.

Guesgen, H. W. (1989). Spatial Reasoning Based on Allen's Temporal Logic. Technical Report TR-89-049, ICSI, Berkeley, Cal.

Hernández, D. (1994). *Qualitative Representation of Spatial Knowledge*. LNAI 804. Springer.

Hofstedt, P. (2000). Cooperating Constraint Solvers. In *Sixth International Conference on Principles and Practice of Constraint Programming - CP*, volume 1894 of *LNCS*. Springer.

Johnson-Laird, P. N. (1983). *Mental Models: Towards a Cognitive Science of Language, Inference and Consciousness.* Cambridge University Press, Cambridge.

Jörding, T. and Wachsmuth, I. (1996). An Antropomorphic Agent for the Use of Spatial Language. In *Proceedings of ECAI'96-Workshop on Representation and Processing of Spatial Expressions*, pages 41–53.

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

Röfer, T. (1997). Routemark-based Navigation of a Wheelchair. In *Third ECPD International Conference on Advanced Robotics, Intelligent Automation and Active Systems*, Bremen.

Schulmeister, B. and Wysotzki, F. (1994). The Piecewise Linear Classifier DIPOL92. In Bergadano, F. and de Raedt, L., editors, *Proceedings of the European Conference on Machine Learning*, LNAI 784, pages 411–414, Berlin. Springer.

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

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

Wiebrock, S. and Wysotzki, F. (1999). Lernen von räumlichen Relationen mit CAL5 und DIPOL. Technical Report 99-17, TU Berlin.