QSRlib: a software library for online acquisition of Qualitative Spatial Relations from Video

Y Gatsoulis[1], M Alomari[1], C Burbridge[3], C Dondrup[2], P Duckworth[1], P Lightbody[2], M Hanheide[2], N Hawes[3], D C Hogg[1], A G Cohn[1]
[1] University of Leeds, UK; [2] University of Lincoln, UK; [3] University of Birmingham, UK.

Abstract

There is increasing interest in using Qualitative Spatial Relations as a formalism to abstract from noisy and large amounts of video data in order to form high level conceptualisations, e.g. of activities present in video. We present a library to support such work. It is compatible with the Robot Operating System (ROS) but can also be used stand alone. A number of QSRs are built in; others can be easily added.

Introduction

Humans can effectively make sense of their surroundings and easily recognise the activities that take place using their perceptual abilities. Although these cognitive abilities are highly complex and not yet fully understood with many theories being suggested (Johnson-Laird 2008), there is general consensus that spatio-temporal relations and reasoning play an important role (Ragni, Fangmeier, and Bruessow 2010). For this, there are dedicated areas in our brains (Amorapanth, Widick, and Chatterjee 2010), which are able to form efficient and categorical qualitative representations of spatial relations resulting in powerful levels of flexible abstractions for inference and reasoning while avoiding information overflow and computational bottlenecks (Laeng 2013).

Within the field of artificial intelligence there has been considerable interest in developing techniques to make intelligent systems with similar spatial reasoning abilities. To this end, many spatial qualitative representations and calculi have been developed (Cohn and Renz 2008; Chen, Cohn, and Liu 2015), and have been used in a number of real-world application domains such as geographical systems (Van de Weghe et al. 2005), video analysis (Sridhar, Cohn, and Hogg 2011a), human activity recognition (Behera, Cohn, and Hogg 2012; Tayyub et al. 2014), etc. Despite their successes, there is no formal methodology to help decide which QSR is best suited for a given task, and often this is determined using domain knowledge, data analysis and empirical experimentation (Sridhar, Cohn, and Hogg 2011b).

As a result of all theoretical work that has taken place in developing new calculi for QSR, there have been a number of implemented systems with the aim of offering generic QSR reasoning services. For example SNARK (Stickel, Waldinger, and Chaudhri 2000) is an automated theorem proving system which builds in the mereotopological calculus, RCC, as does the commense knowledge system, CYC (Grenon 2003). SparQ [1] (Dylla et al. 2006) provides implementations of a wide variety of QSR calculi, as does CLP(QS). Another system providing qualitative spatial reasoning services is the Qualitative Algebra Toolkit (QAT) [2] (Condotta, Ligozat, and Saade 2006) which is a Java based toolkit implementing a constraint based approach for reasoning and has XML definitions of several qualitative algebras. GQR[3] (Westphal, Wölfl, and Gantner 2009) is another constraint based solver which supports binary constraint calculi for qualitative spatial/temporal reasoning.

However, the focus of all these systems is on *symbolic reasoning* – i.e. they generally assume that knoweldge is expressed already in symbolic form as a knowledge base of assertions involving qualitative relations over spatial entities and the reasoning services provided are primarily consistency checking and compositional reasoning; the issue of abstracting from metric data to form the qualitative knowledge base is not addressed at all (CLP(QS) does in fact allow for polygons to be directly input, but the focus is stil on symbolic reasoning). QSRlib is complementary to these systems in that it does not provide symbolic reasoning, but concentrates on qualitative abstraction from metric data. Moreover, these systems are not purposely designed for use on ROS-based robotic platforms, primarily owing to their choice of programming languages.

On the other hand there is one form of symbolic reasoning concerning *conceptual neighbourhoods* which these systems do not provide; a *conceptual neighbourhood* diagram (CND) – see Figure 3 – specifies which relations are conceptual neighbours – i.e. relations $R1$ and $R2$ are conceptual neighbours if $R1(a, b)$ holds and subsequently $R2(a, b)$ holds owing to the entities involved continuously deforming/translating, and there is no intermediate relation $R3(a, b)$ which holds. However, when data is acquired from sensors, and objects may be moving fast relative to the frame speed and relationship granularity, or objects may be occluded, then it is possible that a relation in the sequence

[1] http://www.sfbtr8.uni-bremen.de/project/r3/sparq/
[2] http://www.cril.univ-artois.fr/~saade/QAT/
[3] http://www.sfbtr8.spatial-cognition.de/de/projekte/reasoning/r4-logospace/research-tools/gqr/

through a CND may not be observed. In this case it may be useful to perform reasoning to detect this discrepancy and take corrective action (which might either be to insert the missing relations or to discard the observation, as in (Fernyhough, Cohn, and Hogg 1998)). This service is however provided in QSRlib.

As such, researchers often re-invent the wheel by repeating the implementations of QSRs. In an attempt to resolve these issues and speed up background development allowing better use of research time, we have developed a modern library, named QSRlib[4,5], with the aims of:

- Providing a number of QSRs that are well known, and in common use in scientific community.
- Exposing these QSRs via a standard IO interface that allows quick and easy re-usability, including a ROS interface to allow use in cognitive robotic systems.
- Providing a flexible and easy to use infrastructure to allow rapid development of new QSRs that extend the library.
- Delivering abstracted QSRs over time in an aggregated representation that facilitates further inference.

A typical usage of QSRlib would be an intelligent system, such as a robot for example, which acquires visual data via an RGB-D camera, such as a Kinect, and via object recognition and skeleton tracking is able to perceive the individual entities in the world. The system can then make calls to QSRlib in order to abstract this input data and form a qualitative representation of the perceived world scene. This could then be used to recognise activities in natural scenes such as the one shown in Figure 1, using already learnt models expressed using QSRs in the QSRlib library.

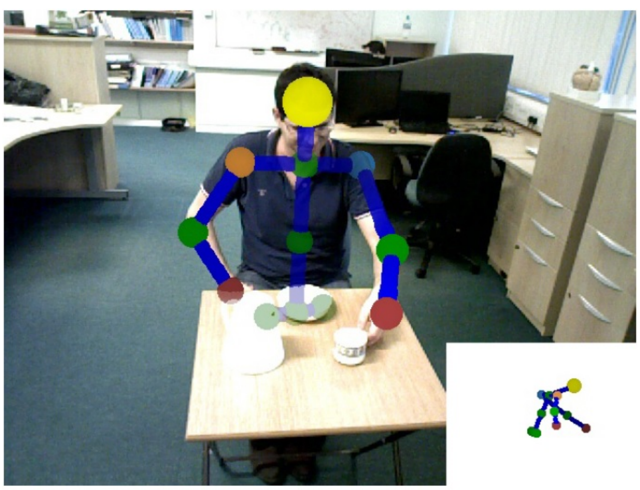

Figure 1: Activity recognition in a table top setting. Dyadic QSR relations between detected objects/skeleton points can be computed (bottom right inset).

[4]http://github.com/strands-project/strands_qsr_lib

[5]http://qsrlib.readthedocs.io

Description

QSRlib is based on a client-server architecture implemented in python 2.7 although measures for compatibility with 3.x have been adopted. Furthermore, the library can also be used with the Robot Operating System[6] (ROS), allowing its use in complex intelligent systems, such as robots. Figure 2 presents a flowchart with the main step processes for computing QSRs via the library. Qualitative spatial relations typically operate on object data such as their Cartesian positions, rotations, edges or bounding boxes describing their shape, velocities, etc., retrieved from single or multiple frames using 2D/3D trackers that are separate from QSRlib, hence allowing the use of state of the art developments in tracking. The raw data needs to be firstly converted into the common input data format of QSRlib, which represents a timeseries of the states of the perceived objects. Utility functions are provided that allow easy conversion of the raw data to this standard input data structure. This input data structure, the names of the requested QSRs to be computed and other options that control their behaviours are used to create a request message to the QSRlib server, which upon computation returns a response message that includes the computed QSRs as an output data structure similar to the input one, i.e. a timeseries of the QSRs between the objects, as well as other information. The QSRs computation is independent from each other, however, multiple QSRs can be requested and be computed in any frame and returned in both a standard data structure (equivalent to list of ground atomic formulas) as well as in a special graph structure, called a QSTAG (see below) which integrates them all into a single structure over a period of time. In our robot systems in real world deployment tests QSRlib was working from input of a 2D/3D vision system operating at over 20 frames per second.

QSRlib currently consists of directional, distance-based, motion-based, topology-based, and combined direction-topology-based QSRs (see also Table 1). Each of these consists of a set of *Jointly Exhaustive and Pairwise Disjoint* (JEPD) relations between the involved objects (typically two objects); i.e. exactly one relation should hold between any tuple of involved objects. We now briefly describe each of the currently implemented QSRs, and give citations to the literature where the formal definitions and semantics of these relations can be found.

Distance-based: A *Qualitative Distance Calculus* (QDC) (Clementini, Felice, and Hernandez 1997) provides qualitative relations based on a set of parameterised labels and distance boundaries, e.g. ‘touch’: $< .5m$, ‘near’: $> 5m$ and ‘far’: $> 5m$. In QSRlib we extend this and also provide a *Probabilistic Qualitative Distance Calculus* (PQDC) in which there are overlaps between the boundaries, and a probabilistic decision mechanism based on a Gaussian model.

Direction-based: *Cardinal Direction* (Frank 1990; 1996) relations specify compass-like directional relations between two objects with respect to their origin. The *Ternary Point Configuration Calculus* (Moratz, Nebel, and Freksa 2003;

[6]www.ros.org

Table 1: Description of supported qualitative spatial relation families

qualitative spatial relation families	type	num of relations / variations	kind of entities
Qualitative Distance Calculus	distance	user specified	2D points
Probabilistic Qualitative Distance Calculus	distance	user specified	2D points
Cardinal Directions	direction	9	2D rectangles
Moving or Stationary	motion	2	2D points
Qualitative Trajectory Calculus	motion	B11: 9, C21: 81	2D points
Rectangle/Block Algebra	topology & direction	169/2197	2D/3D rectangles
Region Connection Calculus	topology	2, 4, 5, 8	2D rectangles
Ternary Point Configuration Calculus	direction	25	2D points

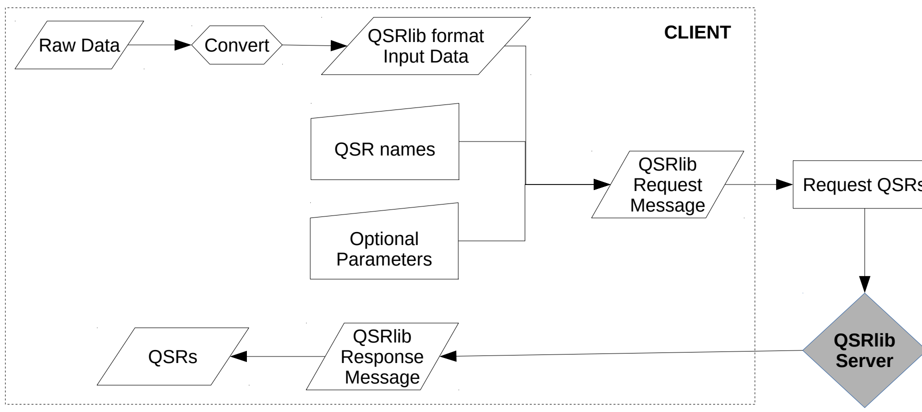

Figure 2: System Architecture.

Dylla and Moratz 2004) is more flexible as it allows the origin to be specified as a parameter, which is an advantage if dealing with multiple frames of reference, and it also further integrates distance-based relations, computed with respect to the horizon defined by the (variable) origin and the relative point of interest.

Topology-based: The *Region Connection Calculus* is a well established calculus for representing and reasoning about the mereotopology of regions. There are different sets of relations depending on the granularity desired, with the most common being RCC-8, which defines eight relations between two regions. These can be seen in Figure 3. The coarser calculus RCC-5 is often more useful in computer vision applications since the tangency distinctions made in RCC-8 but not RCC-5 are unreliable to compute due to noise in low level vision computations. In fact the user is able to supply a *quantisation factor* which allows control of how far apart two regions have to be before they become disconnected. QSRlib implements different variations of RCC, including those shown in Figure 3.

Combined Direction and Topology based: Allen's Interval algebra (Allen 1983) originally put forward as a calculus for qualitative temporal reasoning has also been used for reasoning about space, particularly in its 2D form, the *Rectangle Algebra* wherein objects are projected to the x and y axes and a relation consists of a pair of Allen relations. There are 13 Allen relations (see Figure 4, and hence 13×13=169 Rectangle Algebra relations. There also exists a 3D version of the interval calculus, called the *Block Algebra* (Balbiani, Condotta, and del Cerro 2002) having 13^3 relations which is also implemented in QSRlib.

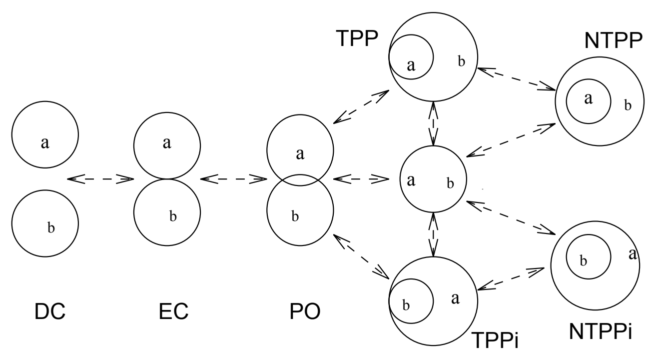

Figure 3: A 2D depiction of RCC-8 relations; RCC-5 is a coarser calculus which collapses DC,EC to DR, TPP, MTPP to PP and TPPi, NTPPi to PPi. The arrows depict the continuous transitions between relations and the entire diagram represents the *conceptual neighbourhood* diagram for RCC8.

Motion-based: The *Qualitative Trajectory Calculus* (QTC) (Van de Weghe et al. 2005; Delafontaine, Cohn, and Van de Weghe 2011) is a calculus for representing relations and reasoning about moving point-objects. There are several variations of QTC which define different types of motion-based relations. For example, QTC-B variants rep-

relation	symbol	inverse	meaning
preceeds	p	pi	
meets	m	mi	
overlaps	o	oi	
starts	s	si	
during	d	di	
finishes	f	fi	
equals	e	e	

Figure 4: The 13 jointly exhaustive and pairwise disjoint Allen interval calculus relations.

resent whether an object is approaching towards or departing from another object, and QTC-C variants compute the relative direction of movement of one object with respect to the other. The full specification of these QTC calculi also allows the specification of whether one object is moving faster, slower, or the same speed as the other. This functionality (present in QTC_{B12} and QTC_{B22}) is currently omitted from QSRlib. The particular versions currently implemented from (Delafontaine, Cohn, and Van de Weghe 2011) in QSRlib are: (1) QTC_{B11} which records the instantaneous motion of two point-objects x and y towards/away from each other: + meaning away from, – means towards and 0 means stationary; since x could be moving towards y but y moving away from x, each relation consists of a pair of these symbols: $\langle\alpha, \beta\rangle$ where $\alpha, \beta \in \{+, 0, -\}$; (2) QTC_{C21} which extends QTC_{B21} by adding two further components forming a quadruple of relations$\langle\alpha, \beta, \gamma, \delta\rangle$ where δ represents whether x is moving to the left (–) or right (+) of the vector $\vec{xy}$ and γ whether y is moving to the left (–) or right (+) of the vector $\vec{yx}$.

Although QTC relations indirectly imply motion, this might not always be the case. For example, QTC_{B21} might compute that one object remains unchanged with respect to another ($\langle 0, 0\rangle$, but this might be the outcome of both objects moving in parallel to each other. For this reason, QSRlib also includes a simple QSR, with just two relations, called *Moving or Stationary*, which as the name implies, determines whether an object is moving or is stationary.

Spatio-temporal relations over timeseries of QSRs For activity recognition we are usually interested in representing the temporal aspects that exist in a timeseries of QSRs. These can be represented in a standard knowledge base of QSR facts and temporal relations between the intervals involved, e.g. (Dubba et al. 2015). An alternative method, which has many advantages for data mining and learning is to use relational graphs, where the edges encode spatial/temporal relations. One particular such representation, known as *Qualitative Spatio-Temporal Activity Graphs* (QSTAG) (Sridhar, Cohn, and Hogg 2010), which provides a compact and efficient graph structure to represent both qualitative spatial and temporal information about objects of interest. In a QSTAG, the temporal relationship between a number of qualitative spatial timepoints is abstracted using Allen's Interval Algebra (Figure 4). A QSTAG has the further advantage of allowing the use of standard graph based methods, such as (approximate) matching. An example QSTAG is shown in Figure 5. QSRlib provides the ability to output QSTAGs abstracted from the input data over a period of time. As can be seen in Figure 5, the layer two spatial nodes can incorporate relations from more than one QSR calculus, in this case QDC (with relations touch/near/...) and QTC_{B21} (with relations $+/0/-$). Thus, QSRlib provides the ability to compute relations from multiple calculi simultaneously and output them in one integrated spatio-temporal representation.

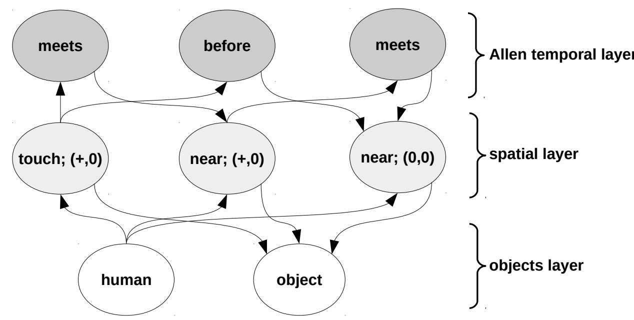

Figure 5: Example of a Qualitative Spatio-Temporal Activity Graph (QSTAG) between a human and an object; each spatial layer node encodes QSRs from two calculi: a QDC relation (touch/near) and a QTC_{B21} one ((+,0)/(0,0)).

Example Usage

QSRlib has been used in various research and teaching projects. We briefly describe some of this usage here to illustrate its possible future uses.

In (Duckworth et al. 2016a) QSRlib was used to rapidly experiment with multiple different types of qualitative representations in order to identify the most suitable one for learning human motion behaviours as perceived by a mobile robot that was deployed for a duration of six weeks in an office environment. QSRlib was used to quickly experiment with suitable representations for classifying scenes and environments from visual data (Thippur et al. 2015; Kunze et al. 2014). The library was also used to compute qualitative relations between a robot and humans moving in order to plan and execute safe path navigation taking into consideration their movement patterns (Dondrup et al. 2015). In (Duckworth et al. 2016b), multiple QSRs were used to represent a detected person's skeleton positions within a semantic map. The QSTAG framework was used to recover latent, semantically meaningful, patterns of how humans move within a scene in an unsupervised setting.

The library has also been used in a number of teaching projects (e.g. recognizing gestures for controlling a device, recognizing someone having breakfast, etc.), allowing the students to concentrate on the more interesting, high level, parts of their projects rather than spending a good portion of their project time in developing the low level tools they need (and which are the same from project to project).

Summary

This paper has presented a software library that allows easy and fast computation of qualitative spatial and temporal relations from objects tracked in video data. A number of implemented systems cited here have employed this library to understand and make predictions about the behaviour of physical systems, even in the presence of noisy quantitative information. Despite the large number of qualitative spatial calculi in the literature, any system aiming to use them to understand video data has had to implement an *ad hoc* solution to abstract the video data to qualitative spatio-temporal relations. QSRlib provides a library to do just this and has implementations of a number of commonly used qualitative calculi.

The standardized I/O data structures of QSRlib allow users to compute and process any set of desired QSRs included in the library with no additional effort. This means that researchers can focus their work on experimenting and analysing with the different types of QSRs, rather than spend time in implementing them and changing the format of their inputs. We have also provided the tools that allow contributors to extend the set of provided QSRs easily, quickly and with flexibility as needed. For example, the online documentation shows a minimal working example of developing and integrating a new QSR to the library; the process contains less than ten lines of key code. QSRlib is open source, well-documented and at a stable state with an active group of developers.

QSRlib is not a system for *reasoning* about qualitative relations, but rather a system for *acquiring* them. Having created a knowledge base of qualitative facts extracted via QSRlib it would be possible then to use the existing complementary QSR reasoning systems such as CLP(QS) or SparQ to perform conventional composition-based reasoning on the extracted qualitative facts. In fact, if data is abstracted from a single source, then it is likely that the extracted representation will be consistent since the world from which it is abstracted will be consistent; of course errors in low level computations and if data is obtained from multiple sources may mean this is no longer the case. Nevertheless, for the case of forming abstracted representations and building models of activity (e.g. learning event models) the principal use of QSRs in the literature has been as a *representation* language, rather than as a *reasoning* mechanism: qualitative spatio-temporal languages are able to abstract away from small metric variations in performance, and from noise in low level visual processing so that activity representations become (more) similar when represented qualitative, which greatly facilitates learning and interpretation of activities in video. QSRlib facilitates the implementation of systems taking this approach to activity.

Acknowledgements

The financial support of the EC under FP7 project STRANDS, 600623, is gratefully acknowledged.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832 – 843.

Amorapanth, P. X.; Widick, P.; and Chatterjee, A. 2010. The neural basis for spatial relations. *Journal of cognitive neuroscience* 22(8):1739 – 53.

Balbiani, P.; Condotta, J.; and del Cerro, L. F. 2002. Tractability results in the block algebra. *J. Log. Comput.* 12(5):885–909.

Behera, A.; Cohn, A. G.; and Hogg, D. C. 2012. Workflow Activity Monitoring using the Dynamics of Pair-wise Qualitative Spatial Relations. In *International Conference on MultiMedia Modeling (MMM), Klagenfurt, Austria*.

Chen, J.; Cohn, A.; and Liu, D. 2015. A survey of qualitative spatial representations. *The Knowledge Engineering Review* 30(1):106136.

Clementini, E.; Felice, P. D.; and Hernandez, D. 1997. Qualitative representation of positional information. *Artificial Intelligence* 95(2):317–356.

Cohn, A., and Renz, J. 2008. *Qualitative Spatial Representation and Reasoning*. Elsevier B.V. chapter 13, 551 – 596.

Condotta, J. F.; Ligozat, G.; and Saade, M. 2006. The QAT: A Qualitative Algebra Toolkit. In *2006 2nd International Conference on Information Communication Technologies*, volume 2, 3433–3438.

Delafontaine, M.; Cohn, A. G.; and Van de Weghe, N. 2011. Implementing a qualitative calculus to analyse moving point objects. *Expert Systems with Applications* 38(5):5187–5196.

Dondrup, C.; Bellotto, N.; Hanheide, M.; Eder, K.; and Leonards, U. 2015. A Computational Model of Human-Robot Spatial Interactions Based on a Qualitative Trajectory Calculus. *Robotics* 4(1):63 – 102.

Dubba, K. S. R.; Cohn, A. G.; Hogg, D. C.; Bhatt, M.; and Dylla, F. 2015. Learning relational event models from video. *Journal of Artificial Intelligence Research* 53:41–115.

Duckworth, P.; Gatsoulis, Y.; Jovan, F.; Hawes, N.; Hogg, D. C.; and Cohn, A. G. 2016a. Unsupervised Learning of Qualitative Motion Behaviours by a Mobile Robot. In *Proc. of the Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'16)*.

Duckworth, P.; Alomari, M.; Gatsoulis, Y.; Hogg, D.; and Cohn, A. 2016b. Unsupervised activity recognition using latent semantic analysis on a mobile robot. In *22nd European Conference on Artificial Intelligence (ECAI) (to appear)*.

Dylla, F., and Moratz, R. 2004. Empirical complexity issues of practical qualitative spatial reasoning about relative position. In *Workshop on Spatial and Temporal Reasoning, ECAI'04*.

Dylla, F.; Frommberger, L.; Wallgrün, J. O.; and Wolter, D. 2006. SparQ: A toolbox for qualitative spatial representation and reasoning. In *Workshop on Qualitative Constraint Calculi: Application and Integration at KI 2006*, 79 – 90.

Fernyhough, J. H.; Cohn, A. G.; and Hogg, D. C. 1998. Building qualitative event models automatically from visual input. In *Proceedings of the International Conference on Computer Vision ICCV98*, 350–355. IEEE.

Frank, A. U. 1990. Qualitative Spatial Reasoning about Cardinal Directions. In Mark, M., and White, D., eds., *Autocarto 10*. Baltimore: ACSM/ASPRS.

Frank, A. U. 1996. Qualitative Spatial Reasoning: Cardinal Directions as an Example. *Geographical Information Systems* 10(3):269–290.

Grenon, P. 2003. Tucking RCC in Cyc's ontological bed. In *IJCAI*, 894–899.

Johnson-Laird, P. 2008. *How We Reason*. Oxford University Press (OUP).

Kunze, L.; Burbridge, C.; Alberti, M.; Thippur, A.; Folkesson, J.; Jensfelt, P.; and Hawes, N. 2014. Combining Top-down Spatial Reasoning and Bottom-up Object Class Recognition for Scene Understanding. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'14)*.

Laeng, B. 2013. *Representation of Spatial Relations*. Oxford University Press.

Moratz, R.; Nebel, B.; and Freksa, C. 2003. Qualitative spatial reasoning about relative position: The tradeoff between strong formal properties and successful reasoning about route graphs. In Freksa, C.; Brauer, W.; Habel, C.; and Wender, K. F., eds., *Lecture Notes in Artificial Intelligence 2685: Spatial Cognition III*. Berlin, Heidelberg: Springer Verlag. 385–400.

Ragni, M.; Fangmeier, T.; and Bruessow, S. 2010. Deductive spatial reasoning: From neurological evidence to a cognitive model. In *Proceedings of the 10th International Conference on Cognitive Modeling*, 193 – 198.

Sridhar, M.; Cohn, A. G.; and Hogg, D. C. 2010. Unsupervised learning of event classes from video. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Sridhar, M.; Cohn, A.; and Hogg, D. 2011a. From Video to RCC8: Exploiting a Distance Based Semantics to Stabilise the Interpretation of Mereotopological Relations. In *Spatial Information Theory - 10th International Conference (COSIT 2011)*, 110 – 125.

Sridhar, M.; Cohn, A. G.; and Hogg, D. C. 2011b. Benchmarking Qualitative Spatial Calculi for Video Activity Analysis. In *Proc. IJCAI Workshop Benchmarks and Applications of Spatial Reasoning*, 15–20.

Stickel, M. E.; Waldinger, R. J.; and Chaudhri, V. K. 2000. A guide to SNARK. Technical report, AI Center, SRI International, Menlo Park, USA.

Tayyub, J.; Tavanai, A.; Gatsoulis, Y.; Cohn, A.; and Hogg, D. 2014. Qualitative and Quantitative Spatio-Temporal Relations in Daily Living Activity Recognition. In *Proc. of the 12th Asian Conference on Computer Vision (ACCV'14)*. Springer.

Thippur, A.; Burbridge, C.; Kunze, L.; Alberti, M.; Folkesson, J.; Jensfelt, P.; and Hawes, N. 2015. A Comparison of Qualitative and Metric Spatial Relation Models for Scene Understanding. In *29th Conference on Artificial Intelligence (AAAI'15)*.

Van de Weghe, N.; Cohn, A.; De Tré, B.; and De Maeyer, P. 2005. A Qualitative Trajectory Calculus as a basis for representing moving objects in Geographical Information Systems. *Control and Cybernetics* 35(1):97–120.

Westphal, M.; Wölfl, S.; and Gantner, Z. 2009. GQR: A fast solver for binary qualitative constraint networks. In *Benchmarking of Qualitative Spatial and Temporal Reasoning Systems, Papers from the 2009 AAAI Spring Symposium, Technical Report SS-09-02, Stanford, California, USA, March 23-25, 2009*, 51–52.