

Robot Control based on Qualitative Representation of Human Trajectories

Nicola Bellotto

School of Computer Science
University of Lincoln
LN6 7TS, United Kingdom

Abstract

A major challenge for future social robots is the high-level interpretation of human motion, and the consequent generation of appropriate robot actions. This paper describes some fundamental steps towards the real-time implementation of a system that allows a mobile robot to transform quantitative information about human trajectories (i.e. coordinates and speed) into qualitative concepts, and from these to generate appropriate control commands. The problem is formulated using a simple version of qualitative trajectory calculus, then solved using an inference engine based on fuzzy temporal logic and situation graph trees. Preliminary results are discussed and future directions of the current research are drawn.

Machine perception and interpretation of human behaviours are perhaps some of the most challenging tasks that research communities in computer vision and robotics have to face with. Solutions in this area are necessary for a number of important applications, from automated video surveillance to assistive robotics for domestic environments (Bellotto et al. 2009; Galindo et al. 2011). A restricted class of human behaviours is concerned with the (usually goal-oriented) motion of an agent, which can be associated with the actions of walking towards something or someone, standing still, etc.. The automatic interpretation of such motion, being through vision or other sensing modalities, is important in particular for mobile robots designed to provide services for humans. For example, a drink-serving robot should be able to detect and track the motion of a potential customer, and infer from his/her behaviour whether he/she is interested in having a drink or not.

There is a vast amount of recent literature proposing solutions for tracking people with mobile robots, either using vision, laser or multi-sensor approaches (Bellotto and Hu 2010; Luber, Tipaldi, and Arras 2011). However, if ever considered, the robot actions that follow human tracking are usually associated with a numerical or topological representation of the current target position, rather than with a semantic interpretation of his/her motion activity. For example, Nakauchi and Simmons (2002) developed a social robot that observes the position and orientation of

people queueing in front of a counter, and then enters the queue at the rear. Bennewitz et al. (2005), instead, proposed a solution to learn typical motion patterns of people in an office environment, using this information to predict the location of a person even when the robot is in a different room. Recent works have also considered the motion activity of people in relation to their spatial location, so that a social robot can predict the position of potential users and proactively approach them (Kanda et al. 2009; Chung and Huang 2010). These solutions, however, use numerical rather than qualitative data to reason about motion activity. Moreover, the proposed models do not incorporate the effect introduced by the robot's actions. To do this, the behaviour of the robot has to be included in the activity model as well.

The approach suggested in this paper adopts a compact and effective method for qualitative representation of human/robot trajectories. Although demonstrated in a simple scenario, a major contribution of this work lies on the scalability of the proposed approach, which makes it feasible for representing long term interactions between multiple agents, and can eventually facilitate robot programming for complex social navigation tasks. Another main contribution is the framework used for the integration of several tools, from the field of AI, computer vision and robotics, to implement a cognitive system for mobile robots interacting with humans. The proposed solution is inspired by previous cognitive vision systems (Nagel 2004) and extends some recent work in automatic video surveillance (Bellotto et al. 2012) to the robotics research field.

Representation of Human Trajectories

In order for a robot to understand human motion and act accordingly, it is necessary to interpret semantically the data collected by its sensors. In other terms, quantitative information about human trajectories have to be converted into qualitative concepts, so that numerical coordinates, speed, direction, etc., of an agent can be represented with actions like “moving towards” or “moving away from”. Qualitative spatial representation and reasoning is an active research area that deals, among the others, with the formalization of spatial relations between physical entities (Cohn and Renz 2008). One such formalization, called Qualitative Trajectory Calculus (QTC), considers in particular the relative motion

between two points in 1- or 2-dimensional spaces (Van de Weghe et al. 2006).

QTC can be used to represent and reason about the relative motion of two agents, the first being a person (identified here as k) and the second a robot (identified as l). Adopting the same notation used in Van de Weghe et al. (2006) for the 1-dimensional case, where only the straight line passing by k and l is considered, the following movements can be identified:

1. movement of k with respect to l at time t
 - $-$: k is moving towards l
 - $+$: k is moving away from l
 - 0 : k is stable with respect to l
2. movement of l with respect to k at time t
 - same as above, but with k and l swapped
3. relative speed of k with respect to l at time t
 - $-$: k is slower than l
 - $+$: k is faster than l
 - 0 : k has the same speed of l

The situation, for example, where “a person k approaches the robot l , which is not moving” can be simply represented in QTC¹ as $(-0+)$. Besides its clarity and compact representation, the adoption of such a formalism will prove particularly useful in future extension of the system, where 2-dimensional relations and their combination in *conceptual neighbourhood diagrams* and *composition tables* (Delafontaine et al. 2011) can be used to generate much more complicated motion patterns. The current implementation is limited to the 1-dimensional case. This allows for basic human-robot interactions with very few rules, enough for the purpose of showing the feasibility of the proposed approach. The speed component has also been ignored, so only the relative movements described at the previous points 1 and 2 are actually considered. The example above would therefore simplify to the relation (-0) .

QTC is generally applicable to continuous spaces, but since in the actual system sensor data is available at constant intervals (i.e. discrete time), the original conditions reduce to the following basic rules:

1. movement of k with respect to l at time t
 - $-$: $d(k_{t-1}, l_t) > d(k_t, l_t) \wedge d(k_t, l_t) > d(k_{t+1}, l_t)$
 - $+$: $d(k_{t-1}, l_t) < d(k_t, l_t) \wedge d(k_t, l_t) < d(k_{t+1}, l_t)$
 - 0 : all the other cases
2. movement of l with respect to k at time t
 - same as above, but with k and l swapped

where $d(k_t, l_t)$ corresponds to the Euclidean distance between k and l at time step t , while $t - 1$ and $t + 1$ are an instant before and an instant later respectively.

These rules can be implemented using the constructs of temporal logic. The current system uses in particular

¹This particular case, where relative 1-dimensional motion and speed are considered, is actually termed QTC_{B2}. If the speed relation is ignored, it is called QTC_{B1} (Delafontaine et al. 2011).

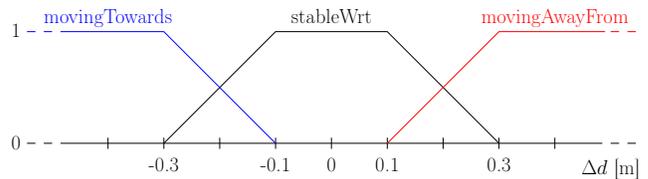


Figure 1: Membership functions (degree of validity) of the three different motion cases.

F-Limette, an inference engine based on Fuzzy Metric-Temporal Horn Logic (FMTHL), which is freely available online² (Gerber and Nagel 2008). Using this language, it is possible to assign a *degree of validity*, which is effectively the application of a trapezoidal membership function, to each one of the three motion cases: *movingTowards* ($-$), *stableWrt* (0) and *movingAwayFrom* ($+$). The distance difference $\Delta d = d(k_{t+1}, l_t) - d(k_{t-1}, l_t)$ is the input space on the abscissa (see Figure 1). The F-Limette code implementing these rules is listed in the appendix.

High-Level Reasoning and Robot Control

While there are situations in which models of human motion can be derived from real observations, in many other cases it is preferable to provide an a-priori representation based on expert knowledge. For the problem at hand, knowledge about particular motion behaviours, expressed in terms of QTC rules, can be encoded within F-Limette using a schematic representation called Situation Graph Tree (SGT) (Nagel 2004). In an SGT (see for example Figure 2), a single *situation* corresponds to a particular agent state, at a specific time instant, that satisfies one or more logic predicates. There can also be associated actions that the agent is expected to carry while in that state. A situation can be temporally connected to another one with a *prediction edge*, which effectively describes “what should happen next”. Situations and prediction edges form a *situation graph*. When organized in a tree structure, situation graphs at the bottom-layer represent more detailed specializations of the situations at the top. During the inference process, the SGT is traversed in a depth-first fashion to find instantiable situations.

In the current implementation, the system includes a simple SGT with two layers, shown in Figure 2, that describes the case of a human moving towards or away from the robot. More interesting and complex behaviours could be modelled using larger multi-layer SGTs – see for example Nagel (2004) and Bellotto et al. (2012). A typical scenario would be the following: “A person k moves towards the robot l , then stops for a moment. In the meanwhile, the robot starts to approach the person. The latter, however, turns back and leaves, so the robot stops”. Using QTC, this could be represented as follows:

$$(-0) \rightsquigarrow (--) \rightsquigarrow (0-) \rightsquigarrow (+-) \rightsquigarrow (+0) \quad (1)$$

The five situations are encoded at the bottom of the SGT, with arrows (i.e. prediction edges) indicating their temporal

²<http://cogvisys.iaks.uni-karlsruhe.de/Vid-Text/>

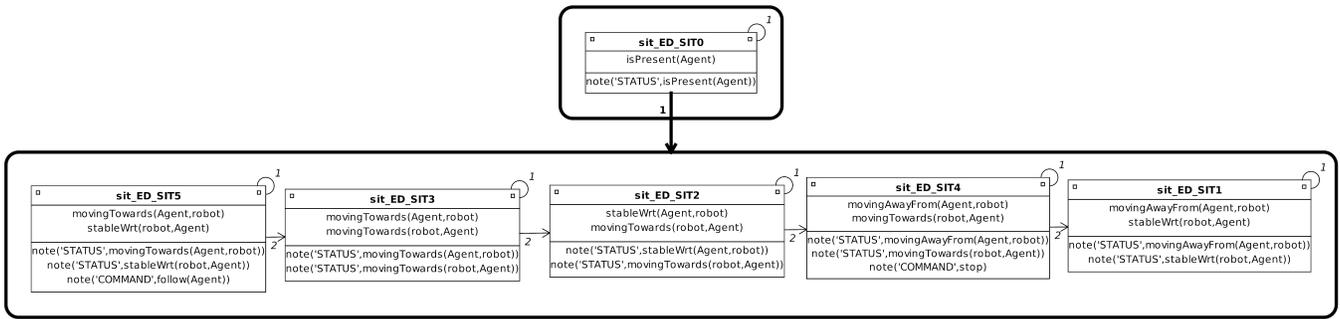


Figure 2: Situation Graph Tree of the human motion behaviour and relative robot actions.

sequence. Whenever new evidence about the current position and speed of a nearby person is available, an inference process is started on F-Limette with an SGT traversal. The traversal at time $t + 1$ starts from the previous successful instantiation of a situation at time t (which is the root if the last traversal had failed).

An important difference of this system compared to previous SGT-based solutions is the integration of high-level commands for controlling the motion of the robot. Every situation in the SGT has one or more actions `note` associated with it, which F-Limette converts in string messages to print and/or send to other sub-systems. In particular, two kind of messages are implemented: a `STATUS` message to describe the current situation, and a `COMMAND` message to control the robot. While `STATUS`-like strings were already used in Nagel (2004) to generate natural language, `COMMAND` strings were only partially exploited in Bellotto et al. (2012) to control a set of pan-tilt cameras and gather more visual information. However, because a camera was never considered as an agent itself, the influence of such high-level commands on the next inference iteration was not fully explored. This is an important novelty of the current system. It is clear from the SGT in Figure 2, indeed, that the action-commands `follow(Agent)` and `stop`, both executed by the `robot`, have a direct effect on the potential instantiation of the next situation. In particular, `follow(Agent)` will enable the transition $(-0) \rightsquigarrow (-)$ in Equation 1, while `stop` will cause $(+-) \rightsquigarrow (+0)$. Although here demonstrated for a relatively simple task, it is argued that the rigorous approach adopted so far could be easily extended to deal with much more complicated scenarios and robot control policies.

Implementation and Results

The solution has been fully implemented, but a systematic experimentation has not been completed as yet, so only anecdotal results are available at this point. Several tests have been performed in a simulated environment³ where moving agents, resembling people, wander around a large indoor environment avoiding obstacles. A mobile robot tracks them using laser-based leg detection and Bayesian filtering (Bellotto and Hu 2010). The smooth nearness-

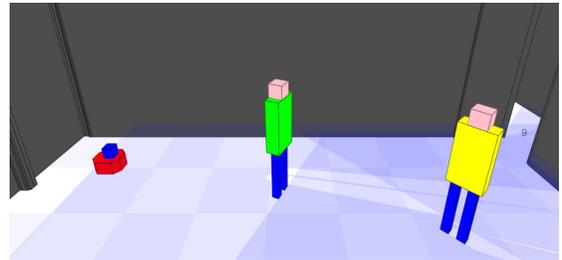


Figure 3: Simulated environment with robot and people.

diagram algorithm is used for navigation and obstacle avoidance (Durham and Bullo 2008). A snapshot of the simulation is shown in Figure 3.

The system architecture is inspired by a previous work (Bellotto et al. 2009), which uses SQL tables as virtual communications channels between three independent components: a laser-based people tracking; a high-level reasoner based on F-Limette; and a control module that converts high-level commands to low-level instructions for the robot⁴. The use of an SQL database is motivated by the possibility to store and retrieve large amounts of information collected during extended periods of time. It simplifies future improvements of the reasoning part to consider long intervals (e.g. hours or days) and makes possible the integration of other systems (e.g. surveillance cameras, robots, etc.) sharing the same information.

In a typical simulation round, the tracking module would provide x - y position and speed of the current agents (i.e. robot and persons) in form of message strings to the SQL server, e.g. `hasStatus(Agent, Xpos, Xvel, Ypos, Yvel)`. These are retrieved by the reasoner and included as new evidence, upon which a new inference process is run. The results of the latter, being either status or command strings, are sent back to the SQL server. Possible commands are finally received by the control module that activate the robot platform accordingly⁵.

⁴This part also makes use of the Player/Stage middleware.

⁵Note that in case of a `follow(Agent)` command, the control module will automatically retrieve the current location of the agent to be approached by the robot, without any further intervention by the reasoner, until a `stop` command is received.

³Player/Stage – <http://playerstage.sourceforge.net/>

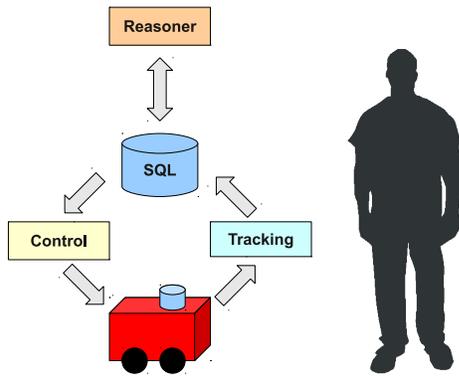


Figure 4: Distributed system architecture.

The solution works in real-time on a quad-core processor with 4GB of RAM, running all the modules of the architecture (including the simulator) except the SQL server. The latter resides on a separate Pentium 4 machine connected via 10/100Mbps LAN. Neither the processing units nor the network have any negative effect on the system performance, the computational requirements of which remain usually very low. Further tests need to be done however with larger sets of FMTHL rules and more complex SGTs.

In general, the system behaves as expected: human movements are correctly interpreted, and the robot commands are properly generated and executed. When a person gets closer, the robot moves in his/her direction as well. If the person turns back though, the robot stops. Nothing happens if the person starts to be tracked while already moving away. In case of several people, the robot acts only according to the movements of the first detected person, at least until the latter is not tracked any more. The simplicity of the 1-dimensional QTC representation shows however its limitations when a person is considered to be approaching the robot only because the distance between the two decreases, even if the person does not point towards the robot at all. As explained in the next section, reasoning with 2-dimensional QTC is a necessary improvement for the implementation of more advanced robot behaviours.

Conclusions and Future Work

The work here presented constitutes the first attempt to implement a complex system for the semantic interpretation of human motion and simultaneous generation of high-level robot commands. The solution is based on a robust and well-tested integration of concepts developed in AI, computer vision and robotics, the aim of which is to contribute in the design of a cognitive robot for assisting humans in daily tasks.

Despite encouraging results, it is clear that a number of proper experiments are necessary to validate the current approach. Simulation tests are of little value when the entities involved are real humans, the behaviour of whom is generally unpredictable. The system will have to be fully evaluated in a real-world environment, and experiments carried out with a range of different people.

There are a number of aspects in which the current system

could be improved and extended. During initial testing, it became clear that a 2-dimensional QTC is the minimal requirement to better represent the variety of human movements. This would consider the direction of an agent with respect to the other one. Also, the potential of QTC still needs to be fully exploited considering speed and using composition tables and other tools to combine motion relations between more than two agents (i.e. robot and several people).

Finally, it would be interesting in the future to analyse the interactions between different agents, in particular the correlation between robot actions and human motion, and to generate high-level commands that guide the robot towards optimal observation and interaction points.

Acknowledgments

The author would like to thank the anonymous reviewers for their valuable comments and suggestions. He is also grateful to former colleagues at the Active Vision Group in Oxford for fruitful discussions about this work.

References

- Bellotto, N., and Hu, H. 2010. Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of Bayesian filters. *Autonomous Robots* 28(4):425–438.
- Bellotto, N.; Sommerlade, E.; Benfold, B.; Bibby, C.; Reid, I.; Roth, D.; Fernández, C.; Van Gool, L.; and González, J. 2009. A distributed camera system for multi-resolution surveillance. In *Proc. of the 3rd ACM/IEEE Int. Conf. on Distributed Smart Cameras (ICDSC)*.
- Bellotto, N.; Benfold, B.; Harland, H.; Nagel, H.-H.; Pirlo, N.; Reid, I.; Sommerlade, E.; and Zhao, C. 2012. Cognitive visual tracking and camera control. *Computer Vision and Image Understanding* 116(3):457–471.
- Bennewitz, M.; Burgard, W.; Cielniak, G.; and Thrun, S. 2005. Learning motion patterns of people for compliant robot motion. *Int. Journ. of Robotics Research* 24(1):31–48.
- Chung, S.-Y., and Huang, H.-P. 2010. A mobile robot that understands pedestrian spatial behaviors. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 5861–5866.
- Cohn, A. G., and Renz, J. 2008. Chapter 13 Qualitative Spatial Representation and Reasoning. In van Harmelen, F.; Lifschitz, V.; and Porter, B., eds., *Handbook of Knowledge Representation*, volume 3. Elsevier. 551–596.
- Delafontaine, M.; Chavoshi, H.; Cohn, A.; and Van de Weghe, N. 2011. A qualitative trajectory calculus to reason about moving point objects. In *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions*. IGI Global, Hershey, PA, USA.
- Durham, J., and Bullo, F. 2008. Smooth Nearness-Diagram Navigation. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 690–695.
- Galindo, C.; González, J.; Fernández-Madrigal, J.-A.; and Saffiotti, A. 2011. Robots that change their world: Inferring goals from semantic knowledge. In *Proc. of the 5th European Conference on Mobile Robots (ECMR)*, 1–6.

Gerber, R., and Nagel, H.-H. 2008. Representation of occurrences for road vehicle traffic. *Artificial Intelligence* 172(4-5):351–391.

Kanda, T.; Glas, D. F.; Shiomi, M. F.; and Hagita, N. F. 2009. Abstracting people’s trajectories for social robots to proactively approach customers. *IEEE Trans. on Robotics* 25(6):1382–1396.

Luber, M.; Tipaldi, G.; and Arras, K. 2011. Place-dependent people tracking. *International Journal of Robotics Research* 30(3):280–293.

Nagel, H.-H. 2004. Steps toward a cognitive vision system. *AI Magazine* 25(2):31–50.

Nakauchi, Y., and Simmons, R. 2002. A social robot that stands in line. *Autonomous Robots* 12(3):313–324.

Van de Weghe, N.; Cohn, A. G.; Tré, G. D.; and Maeyer, P. D. 2006. A qualitative trajectory calculus as a basis for representing moving objects in geographical information systems. *Control and Cybernetics* 35(1):97–119.

Appendix – *F-Limette* Implementation of QTC

```
always (isPresent(Agent) :-
  Agent <> robot,
  prev hasStatus(Agent,_,_,_),
  hasStatus(Agent,_,_,_),
  next hasStatus(Agent,_,_,_)
).

always ( dist([Xa, Ya], [Xb, Yb], D) :-
  D is ((Xb-Xa) * (Xb-Xa) + (Yb-Ya) * (Yb-Ya)) ^ 0.5
).

always ( distance(K, L, D) :-
  K <> L,
  hasStatus(K, Xk, _, Yk, _),
  hasStatus(L, Xl, _, Yl, _),
  dist([Xk, Yk], [Xl, Yl], D)
).

always ( prevDistance(K, L, D) :-
  K <> L,
  prev hasStatus(K, Xk, _, Yk, _),
  hasStatus(L, Xl, _, Yl, _),
  dist([Xk, Yk], [Xl, Yl], D)
).

always ( nextDistance(K, L, D) :-
  K <> L,
  next hasStatus(K, Xk, _, Yk, _),
  hasStatus(L, Xl, _, Yl, _),
  dist([Xk, Yk], [Xl, Yl], D)
).

always( movingTowards(K, L) :-
  prevDistance(K, L, Dprev),
  distance (K, L, D),
  nextDistance(K, L, Dnext),
  Dprev > D,
  D > Dnext,
  degreeOfValidity(Dnext-Dprev, -10000, -9999, -0.3, -0.1)
).

always( movingAwayFrom(K, L) :-
  prevDistance(K, L, Dprev),
  distance (K, L, D),
  nextDistance(K, L, Dnext),
  Dprev < D,
  D < Dnext,
  degreeOfValidity(Dnext-Dprev, 0.1, 0.3, 9999, 10000)
).

always( stableWrt(K, L) :-
  prevDistance(K, L, Dprev),
  distance (K, L, D),
  nextDistance(K, L, Dnext),
  not (Dprev > D, D > Dnext; Dprev < D, D < Dnext),
  degreeOfValidity(Dnext-Dprev, -0.3, -0.1, 0.1, 0.3)
).
```