From Sequence to Trajectory and Vice Versa: Solving the Inverse QTC Problem and Coping with Real-World Trajectories

Konstantinos Iliopoulos
NCSR Demokritos
Agia Paraskevi, Greece
helix@fulbrightmail.org

Nicola Bellotto
University of Lincoln
Lincoln, United Kingdom
nbellotto@lincoln.ac.uk

Nikolaos Mavridis
NCSR Demokritos
Agia Paraskevi, Greece
nmav@alum.mit.edu

Abstract

Spatial interactions between agents carry information of high value to human observers, as exemplified by the high-level interpretations that humans make when watching the Heider and Simmel movie, or other such videos which just contain motions of simple objects, such as points, lines and triangles. However, not all the information contained in a pair of continuous trajectories is important; and thus the need for qualitative descriptions of interaction trajectories arises. Towards that purpose, Qualitative Trajectory Calculus (QTC) has been proposed in (Van de Weghe 2004). However, the original definition of QTC handles uncorrupted continuous-time trajectories, while real-world signals are noisy and sampled in discrete-time. Also, although QTC presents a method for transforming trajectories to qualitative descriptions, the inverse problem has not yet been studied. Thus, in this paper, after discussing several aspects of the transition from ideal QTC to discrete-time noisy QTC, we introduce a novel algorithm for solving the QTC inverse problem; i.e. transforming qualitative descriptions to archetypal trajectories that satisfy them. Both of these problems are particularly important for the successful application of qualitative trajectory calculus to Human-Robot Interaction.

Introduction

As the epitome of the philosophy of Heraclitus (544-484BC) states: "All entities move and nothing remains still". Thus change, and especially motion (which is the primary sensory manifestation of change), are central elements in almost all philosophical-conceptual systems.

One of the most important species of motion is relative motion between two entities; which forms an essential aspect of spatial interaction, for the case of objects construed as agents (humans, animals, or machines). Such spatial interactions between agents carry information of high value to human observers, as exemplified by the high-level interpretations and judgments that humans make when watching the Heider and Simmel movie (Heider and Simmel 1944), or by the rich semantic content of moving point abstractions of real-world sport or everyday interaction scenes (e.g. reading gender from gait, (Mather and Murdoch 1994)).

Furthermore, such spatial interactions between agents carry invaluable information not only to human observers; but increasingly also to electronic sensing systems, for example those overlooking or assisting with crowd flows (Zhan et al. 2008) or surveillance systems (Bellotto et al. 2012). In recent years, geographical information scientists have intensively explored the topological relationships between multiple moving point objects (MPOs). Research in this area has predominantly focused on the comparison of quantitative characteristics of trajectories such as azimuth, velocity, turning angle, acceleration, and sinuosity. However, when observing the relative motion between two agents, not all the information contained in a pair of continuous trajectories is important. For example, one might not really need the exact distance between two agents; but only the trend of change of relative distance or pose between them. Thus, the need for qualitative descriptions of interaction trajectories arises, abstracting unnecessarily complex complete quantitative representations. One could imagine having an adaptive representation of spatial trajectories of pairs or groups of objects, which can retain exactly as much qualitative information as needed for each application, and which can also be used for learning and reproducing interactive behaviors.

Qualitative Trajectory Calculus (QTC), devised by (Van de Weghe 2004), is a promising development towards this goal. One important feature that differentiates QTC from other qualitative representations is the description of relative rather than absolute motion between two agents, which makes it particularly suitable for describing interaction scenarios independently from the context where they took place. A number of variants of QTC have been proposed in the past, including versions enabling the application of QTC to networks (Delafontaine et al. 2008) and shapes (Van de Weghe et al. 2005). Furthermore, QTC has been applied towards various interaction domains, most importantly including analysis (Hanheide, Peters, and Bellotto 2012) and generation (Bellotto 2012); (Bellotto, Hanheide, and Van de Weghe 2013) of Human Robot Spatial Interactions (HRSI). However, in these papers, the generation of robot behaviors was hand-crafted, and was covering only a number of simple and special cases. Thus, the general inverse problem, which we shall call the QTC reconstruction problem, i.e. how to go from a given QTC sequence to a trajectory pair that satisfies it, was neither defined nor touched upon. Solving the inverse

problem is an essential prerequisite in order to be able to use QTC not only for the analysis, but also for the synthesis of trajectories. Such trajectories, as we shall see, could be generated by robots coming into interaction with other robots or with humans. More specifically, the numerical reconstruction of trajectories from their qualitative descriptions has become increasingly important to create computational models of HRSI, in particular to facilitate the design and implementation of effective robot motion behaviors in a social context (Bellotto, Hanheide, and Van de Weghe 2013). Finally, no clear definitions and guidelines were given on how to analyze noisy and discrete-time-samples real-world measurements of trajectories into meaningful QTC sequences.

In this paper, we shall thus define the QTC reconstruction problem, and provide an algorithm that solves it. We shall also give definitions for discrete-time QTC, and discuss guidelines for several peculiarities that arise in this case as compared to idealized and clean continuous-time trajectories. These include the phenomenon of abrupt transitions, the need for defining thresholds in order to cope with noise, and the special symbolic and syntactical constraints of noisy discrete-time QTC. The solutions given aim to enable the wide-spread application of QTC to complex robot-robot and human-robot interaction problem, moving beyond the handcrafted work of previous papers.

The structure of the paper is as follows: We will start by providing definitions for uni- and bi-directional discrete-time QTC. Then, we will discuss abrupt transitions and thresholding. Later, we shall explore the symbolic and syntactic constraints that hold for zero-threshold as well as non-zero threshold discrete-time QTC. Most importantly, we will then present our proposed QTC reconstruction algorithm, which generates trajectory pairs from given QTC sequences. Finally, following a discussion, we will conclude the paper.

Real-World Discrete-Time QTC

In this section we start with an overview of the original continuous-time QTC and then propose our definitions for uni- and bi-directional discrete-time QTC:

Brief overview of continuous QTC

The following is a brief qualitative overview of what symbols are used in continuous-time QTC. For more details and exact formulas the reader is referred to (Delafontaine et al. 2011).

(A) Distance constraint for the first object, conventionally named k. $'-'$ means that it is approaching the second object, named l, $'+'$ means that is is moving further away, and $'0'$ means that its distance remains steady.

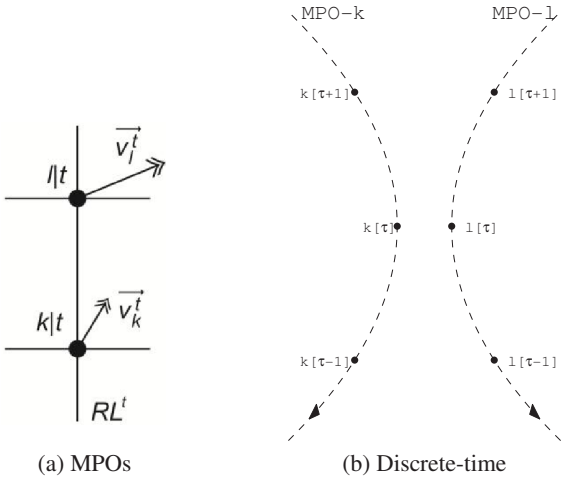
(B) Similar to A but with the roles of k and l interchanged.

(C) Speed constraint; because of the dual nature we only need one such constraint. $'-'$ means that object k is slower than l, $'+'$ that k is faster than l, and $'0'$ that they move with the same speed.

(D) Side constraint for k with respect to line kl: $'-'$ means that k is moving to the left of the line, $'+'$ means that k is moving to the right of the line, and $'0'$ that it moves along the line.

(E) Similar to D but with the roles of k and l interchanged.

(F) Angle constraint: define as ϑ_1 the minimal angle between the velocity vector of k and vector kl and ϑ_2 the equivalent for l. Thus we obtain $'-'$ if $\vartheta_1 < \vartheta_2$, $'+'$ if $\vartheta_1 > \vartheta_2$, and $'0'$ otherwise.



(a) MPOs (b) Discrete-time

Figure 1: Example of MPOs, in this case having QTC_{C2} relation $(-,+,-,+,-,-)$, and two discrete trajectories.

Combinations of the above symbols yield the following QTC variants:

- QTC Basic
  - QTC_{B1}: uses properties [A,B]
  - QTC_{B2}: uses properties [A,B,C]
- QTC Double Cross
  - QTC_{C1}: uses properties [A,B,D,E]
  - QTC_{C2}: uses properties [A,B,C,D,E,F]

Directionality

For the discrete-time case QTC, we have two ways to acquire the symbols that are holding for each time frame. One would be to attempt to mimic the bidirectional paradigm of continuous QTC, by effectively looking at the position of the MPOs during the previous and following frames. This effectively means that we utilize $\{k[\tau-1], k[\tau], k[t+1], l[\tau-1], l[\tau], l[\tau+1]\}$ of fig. 1b. The alternative (unidirectional) is to only examine the current and following positions, essentially attributing the symbols to the "intention of relative movement" of the MPOs. This effectively means that we only utilize $\{k[\tau], k[t+1], l[\tau], l[\tau+1]\}$ of the same figure. Both modes have their own merits, but when we discuss the reconstruction algorithm we will be using the bidirectional mode, simply because it resembles the continuous QTC case the most. We next present the formulas for the discrete case analysis for both modes.

Assume MPOs k and l, and time point τ (See fig. 1a)

$k|\tau$ denotes the position of an MPO k at τ

$d(u, v)$ denotes the Euclidean distance between two positions u and v

$\vec{v_k^\tau}$ denotes the velocity vector of k at τ

RL^τ denotes the reference line through $k|\tau$ and $l|\tau$

$MAA(\vec{v_k^\tau}, RL^\tau)$ denotes the minimum absolute angle between $\vec{v_k^\tau}$ and RL^τ

Bi-directional discrete QTC

(A) Movement of k w.r.t. l at t (distance constraint):

$'-'$**:** k is moving towards l:

$$d(k|\tau-1, l|\tau) > d(k|\tau, l|\tau) > d(k|\tau+1, l|\tau) \quad (1)$$

$'+'$**:** k is moving away from l:

$$d(k|\tau-1, l|\tau) < d(k|\tau, l|\tau) < d(k|\tau+1, l|\tau) \quad (2)$$

$'0'$**:** all other cases

(B) Movement of l w.r.t. k at τ (distance constraint), can be described as in A with k and l interchanged

(C) Relative speed of k w.r.t. l at τ (speed constraint):

$'-'$**:** k is moving slower than l:

$$\left|\vec{v_k^\tau}\right| < \left|\vec{v_l^\tau}\right| \quad (3)$$

$'+'$**:** k is moving faster than l:

$$\left|\vec{v_k^\tau}\right| > \left|\vec{v_l^\tau}\right| \quad (4)$$

$'0'$**:** k and l are moving equally fast:

$$\left|\vec{v_k^\tau}\right| = \left|\vec{v_l^\tau}\right| \quad (5)$$

(D) Movement of k w.r.t. RL^τ (side constraint)

$'-'$**:** k is moving to the left side of RL^τ:

$$k \text{ is on the right side of } RL^\tau \text{ at } \tau-1 \wedge \\ k \text{ is on the left side of } RL^\tau \text{ at } \tau+1 \quad (6)$$

$'+'$**:** k is moving to the right side of RL^τ:

$$k \text{ is on the left side of } RL^\tau \text{ at } \tau-1 \wedge \\ k \text{ is on the right side of } RL^\tau \text{ at } \tau+1 \quad (7)$$

$'0'$**:** all other cases

(E) Movement of l w.r.t. RL^τ (side constraint), can be described as in D with k and l interchanged.

(F) Angle constraint, where:

$'-'$: $\quad MAA(\vec{v_k^\tau}, RL^\tau) < MAA(\vec{v_l^\tau}, RL^\tau) \quad (8)$

$'+'$: $\quad MAA(\vec{v_k^\tau}, RL^\tau) > MAA(\vec{v_l^\tau}, RL^\tau) \quad (9)$

$'0'$: $\quad$ all other cases

Uni-directional discrete QTC

In this case, symbols C and F remain the same except for the fact as in the Bi-directional case, with the only difference that $\vec{v_k^\tau}$ now is defined as $d(k|\tau+1, k|\tau)$. As for symbols A (and B) as well as D (and E), they become:

(A) Movement of k w.r.t. l at t (distance constraint):

$'-'$**:** k is moving towards l:

$$d(k|\tau, l|\tau) > d(k|\tau+1, l|\tau) \quad (10)$$

$'+'$**:** k is moving away from l:

$$d(k|\tau, l|\tau) < d(k|\tau+1, l|\tau) \quad (11)$$

$'0'$**:** all other cases. This, for example, can refer to both k and l being static, or just one of them being static and the other moving in a circular motion around it

(D) Movement of k w.r.t. RL^τ (side constraint):

$'-'$**:** k is moving to the left side of RL^τ:

$$k \text{ is on the left side of } RL^\tau \text{ at } \tau+1 \quad (12)$$

$'+'$**:** k is moving to the right side of RL^τ:

$$k \text{ is on the right side of } RL^\tau \text{ at } \tau+1 \quad (13)$$

$'0'$**:** all other cases

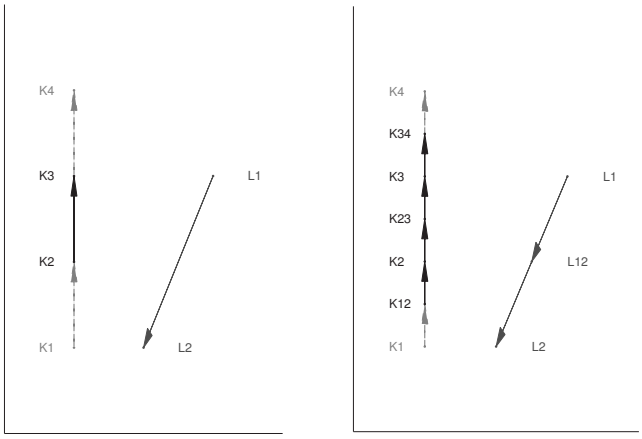
Abrupt transitions

The original definition of QTC suggest that abrupt transitions from $'+'$ to $'-'$ without passing through a $'0'$ are prohibited, and the same prohibition holds for the case of transitions from $'-'$ to $'+'$. This is depicted in the Neighborhood Diagrams (CND) that can be found in (Van de Weghe 2004). However, this constraint does not hold when we move from continuous-time QTC to discrete-time QTC. If one imagines the discrete-time trajectory to have arisen out of a time-sampling of an underlying continuous trajectory, then there is the possibility that the discrete sampling grid will not fall exactly on a time instant that corresponds to a $'0'$ QTC symbol arising out of the continuous trajectory. For example, consider the situation of fig. 2a. For the continuous case, there does exist an interval of infinitesimal length for which we get the $'0'$ symbol for distance, in between the $'-'$ and the $'+'$. For the discrete case though, we get:

$$d(k_1, l_2) > d(k_2, l_2) > d(k_3, l_2) \Rightarrow' -' \\ d(k_2, l_3) < d(k_3, l_3) < d(k_4, l_3) \Rightarrow' +'$$

Furthermore, one should notice that supersampling the already given discrete trajectories does not automatically alleviate the issue of having abrupt transitions. For an example, see figure fig. 2b, which introduces one intermediate point per pair of consecutive points, effectively doubling the length of our trajectories. We still suffer from abrupt transitions, however:

$$d(k_{12}, l_2) > d(k_2, l_2) > d(k_{23}, l_2) \Rightarrow' -' \\ d(k_2, l_{23}) < d(k_{23}, l_{23}) < d(k_3, l_{23}) \Rightarrow' +' \\ d(k_{23}, l_3) < d(k_3, l_3) < d(k_{34}, l_3) \Rightarrow' +'$$



(a) Original. (b) Interpolated.

Figure 2: Abrupt transitions in discrete QTC.

In fact, one might opt for a huge, but finite, number of supersampled intermediate points and never be certain that a $'0'$ will inserted between a $'+'$ and a $'-'$. Of course, if the supersampling is infinite, then we have effectively transitioned to the continuous case again. Conversely, through supersampling, we might accidentally insert $'0'$ between otherwise consecutive $'+'$ which, albeit legitimate, is far from ideal. Thus, abrupt transitions are unavoidable in discrete-time QTC.

Thresholding

In real world situations, most often apart from time-sampling (discrete QTC) there is also noise in our trajectory measurements. The problem is that small perturbations in the positions of the MPOs may greatly affect the exported QTC symbols. As an example, consider the cases where two objects would be moving with the same speed. Clearly, even the slightest noise will cause change the $'0'$ symbol for the speed constraint to become either $'+'$ or $'-'$ and this is unacceptable. Similarly for any other kind of constraints, and most emininently during the production of $'0'$ symbols, we see that noise is quite detrimental for the production of clean symbols, because of the way that the equations are constructed.

Thus it is very important to define thresholds around zero for the various comparisons that have to be satisfied before we may obtain either a $'+'$ or a $'-'$ as a QTC symbol. The question now becomes how to set these thresholds. Note that because of the nature of the equations and the calculations that they imply (euclidean distances for the distance contraint, cross-products for the side constraints etc) there is no way to define a meaningful universal threshold for all constraints. We rather have to define four different threhsolds, one for each of the four types of constraints: distance, speed, side, and angle.

If we can model the statistical behavior of the noise we are dealing with, we can attempt to fine-tune the thresholds accordingly (analytically or empirically). As an idealized example of empirically fine-tuning the thresholds, consider the following: assume that we have a pair of clean, noise-free trajectories which we use to derive clean QTC sequences. Then assume that we can measure the same trajectories but in their noisy version and again derive the corresponding 'dirty' QTC sequences. In formal terms:

$$QTC(s[\tau]) = \text{clean}[\tau] \tag{14}$$

$$QTC(s[\tau] + n[\tau]) = \text{dirty}[\tau] \tag{15}$$

where eq. (15) refers to the case of additive noise.

These two sequences will usually not be identical. In order to compare them, we can use any of the QTC distance metrics introduced in (Van de Weghe 2004). The above two sequences come from unthresholded QTC. However, one can use an appropriate threshold T to derive:

$$QTC_T(s[\tau] + n[\tau]) = \text{thresholded}[\tau] \tag{16}$$

One way to determine an appropriate threshold is to find the threshold T that minimizes the error, i.e.:

$$T = \arg\min_T(QTC_T(s[\tau] + n[\tau]), \text{clean}[\tau]) \tag{17}$$

for an appropriately chosen distance function. Note that the distance function might give different weights to different kinds of substitution errors, depending on the application.

Symbolic and syntactical constraints in discrete-time QTC

For the case of continuous-time QTC_{C2}, not all possible symbol combinations are allowable. As per (Delafontaine et al. 2011) out of the 729 possible combinations of QTC_{C2}, only 305 are possible for the case of two dimensions, due to the interdependence of symbols. Furthermore, not all transitions are possible (for example, as already mentioned, 'abrupt' transitions are disallowed). The question is whether the same holds for the case of discrete-time QTC_{C2}. As we have discussed above, in discrete-time QTC_{C2} transitions from a $'+'$ to a $'-'$ are allowed. But are the same 305 possible combinations allowed in discrete-time QTC_{C2} as they are in continuous-time? To investigate this question we have set out the following experiments. A brownian motion-like trajectory pair was generated through the following iterative process:

$$\vec{s_k}[0] = [-1, 0] \tag{18}$$

$$\vec{s_k}[\tau + 1] = \vec{s_k}[\tau] + r_k[\tau]\,[\cos(\theta_k[\tau]), \sin(\theta_k[\tau])] \tag{19}$$

$$\vec{s_l}[0] = [1, 0] \tag{20}$$

$$\vec{s_l}[\tau + 1] = \vec{s_l}[\tau] + r_l[\tau]\,[\cos(\theta_l[\tau]), \sin(\theta_l[\tau])] \tag{21}$$

where $r_k[\tau], r_l[\tau]$ are random variables with uniform distributions in $[0, 1]$ and $\theta_k[\tau], \theta_l[\tau]$ are also random variables with uniform distribution in $[0, 2\pi]$. For a demonstration, a pair of five-frame long trajectories for brownian motion of k and l can be seen in fig. 3 and the exact coordinates and resulting symbols can be seen in table 1. Note that because we use bidirectional mode it doesn't make sense to define symbols for the first and last frames. Also note that in this case, zero thresholds have been used (see subsection *Thresholding*).

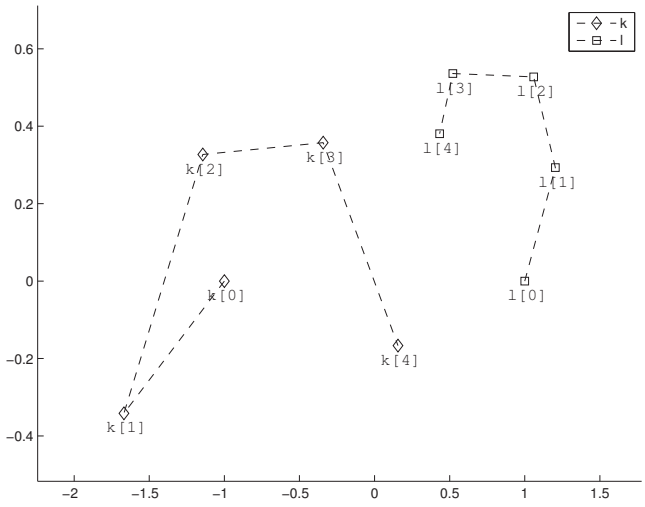


Figure 3: Five-frame long Brownian motion for two MPOs.

k	l	QTC_{C2}
(-1.00, 0.00)	(1.00, 0.00)	Not Defined
(-1.67, -0.34)	(1.20, 0.29)	{0, 0, +, 0, +, -}
(-1.14, 0.33)	(1.06, 0.52)	{-, -, +, 0, +, -}
(-0.34, 0.36)	(0.52, 0.24)	{-, -, +, +, 0, +}
(0.16, -0.17)	(0.43, 0.38)	Not Defined

Table 1: Coordinates and the resulting QTC_{C2} symbols for the brownian motion of fig. 3

Then the corresponding QTC_{C2} sequence was generated out of the trajectory pair $\vec{s_k}[\tau]$, $\vec{s_l}[\tau]$ and the histogram of symbols comprising the QTC sequence was generated. This process was repeated multiple times. In more detail, datasets of increasing size were created for sizes up to 10 million symbols, with 'symbol' meaning an individual 6-tuple $(s_A, s_B, s_C, s_D, s_E, s_F)$ with each element corresponding to a QTC_{C2} constraint. The number of distinct symbols increases with time frame until eventually it converges at approximately time frame 5000 at 288.[1] This is depicted in fig. 4 where one can see the minimum, maximum, average and average $\pm$ 3 standard deviations number of distinct symbols for 100 distinct repetitions of the experiment.

Symbolic constraints for non-zero threshold discrete-time QTC

In the previous subsection we investigated the number of distinct symbols for the case of zero-threshold discrete-time QTC. One question that arises is: is the situation the same for non-zero threshold settings? In order to investigate this question we created the following experiment.

Once again, trajectories were generated according to eqs. (18)–(21). We then calculated the resulting QTC symbols for 10 different sets of thresholds, that can be seen in table 2. In order to get a meaningful initial setting for non-zero thresholds, first we investigated the following related question: what is the threshold setting that will maximize the entropy of the resulting symbol distribution, i.e. distribute the output symbols as uniformly as possible. Entropy is defined as per usual:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{22}$$

The experiments were run for a big number of times ($\#_{repetitions} = 10000$) each for relatively lengthy pairs of trajectories ($\#_{frames} = 10000$). In terms of individual distributions of symbols, we have found that values presented in the the 6th row of table 2 are the ones that maximize the corresponding individual entropies. In fact, for the cases of the velocity ($'C'$) and angular ($'F'$) constraints, we get as close to uniformity as possible. As we keep increasing the thresholds however, we also get more and more $'0'$s, until eventually all $'-'$s and $'+'$s disappear and the entropy becomes zero. The results for the combinations of symbols (rather than individual ones) can be seen in fig. 5.

[1] And seems to remain there at least up to the 10 millionth symbol that we investigated.

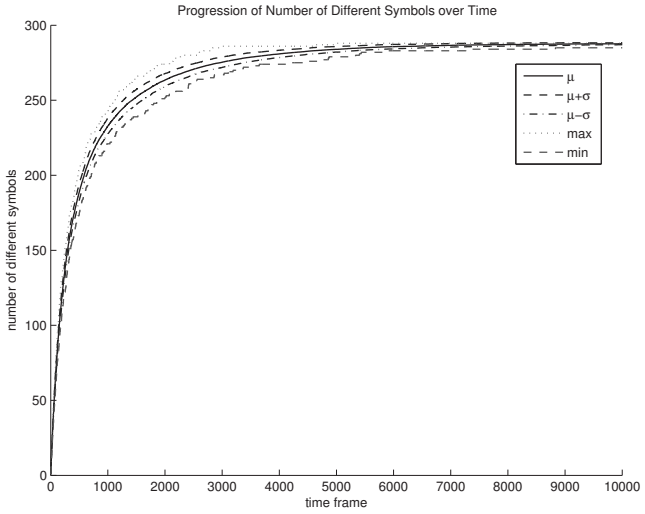


Figure 4: Progression of Number of Different Symbols. We observe that the variance is quite small and that we quickly converge to the value 288.

The Reconstruction Problem

The problem of reconstruction involves utilizing a sequence of some QTC variation as input, and producing an artificial pair of trajectories for two MPOs k and l that satisfy the input. Obviously, many different trajectories can result in the same QTC sequence and thus the inverse problem often has many acceptable solutions. The high-level steps for our proposed reconstruction algorithm are presented next.

Proposed Algorithm

The pseudocode for the reconstruction algorithm is shown in alg. 1. Note that:

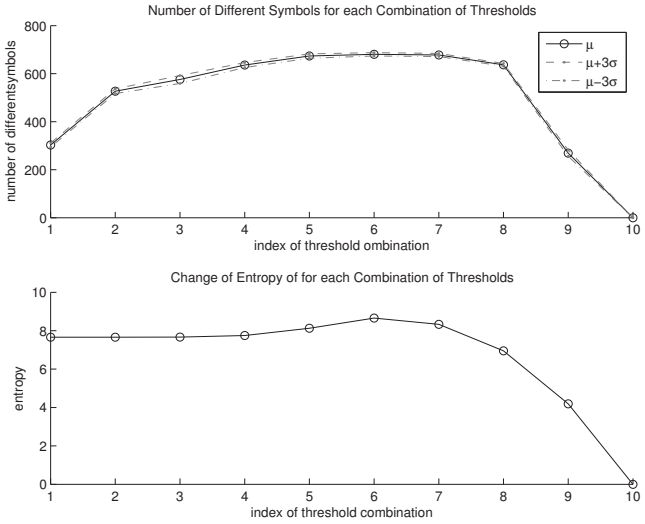


Figure 5: Combinations of thresholds and their effect on number of different symbols & entropy.

Index	Distance	Velocity	Side	Angle
1	0	0	0	0
2	1e-7	1e-5	1e-7	$\pi/18000$
3	1e-6	1e-4	1e-6	$\pi/1800$
4	1e-5	0.001	1e-5	$\pi/180$
5	1e-4	0.01	1e-4	$\pi/18$
6	0.001	0.128	0.001	0.575
7	0.01	0.25	0.1	$\pi/10$
8	0.05	0.5	1	$\pi/5$
9	0.1	0.75	10	$\pi/2$
10	1	1	100	π

Table 2: Thresholds for all four types of Constraints

- Regarding the initial placing of the points, it can be done completely arbitrary (for example, have k be placed at $(-1, 0)$ and l be placed at $(+1, 0)$)
- As already mentioned, this algorithm refers to the bi-directional case. This is why when attempting to match the target tuple of symbols at $\tau - 1$ we are checking both the previous and next location of each MPO
- The way how the repeating step of the algorithm is defined makes this a full backtracking algorithm
- The fact that we choose the objects to remain fixed during the first and last frame enforces a certain QTC vector for frames $i = 2$ and $i = N - 1$, i.e. that they are just $'0's$, but this is not the interesting portion of the trajectories anyway and can be disregarded

What remains to be explained is the way of selection of the candidate points, which is also the most interesting part of the algorithm. We have to keep in mind that we should aim for a strategy that produces the least amount of candidates, else we might easily get stuck in something similar to a local minimum and then have a very hard time to revert to a trajectory that is allowing us to solve the problem. In other words, getting as close as possible to a minimally sufficient grid (set) of candidates is much desired.

Algorithm 1 High-level reconstruction algorithm

$\mathcal{S} \leftarrow$ EMPTY_STACK
$(k_1, l_1) \leftarrow$ SELECT_STARTING_POINTS
$(k_2, l_2) \leftarrow (k_1, l_1)$
for $\tau \in \{3, \ldots, N-1\}$ **do**
  candidate_points $\leftarrow$ FIND_CANDIDATE_POINTS
  repeat
    $(k_\tau, l_\tau) \leftarrow$ DEQUEUE(candidate_points)
  until QTC$(k_{\tau-2}, l_{\tau-2}, k_{\tau-1}, l_{\tau-1}, k_\tau, l_\tau) =$
    $=$ QTC_TARGET$[\tau - 1]$
  if candidate_points $\neq \emptyset$ **then**
    PUSH($\mathcal{S}$, candidate_points)
    INCREMENT(τ)
  else
    candidate_points $\leftarrow$ POP($\mathcal{S}$)
    DECREMENT(τ)
  end if
end for
$(k_N, l_N) \leftarrow (k_{N-1}, l_{N-1})$

We have experimented with various kinds of sets, by combining the following properties: form of the grid (rectangular vs. circular); density of candidates, and orientation (fixed grids vs. grids rotated around the reference line that connects $k|\tau$ and $l|\tau$). The most successful grids (in terms of both a successful and speedy convergence) can be seen in fig. 6. We want to make the following notes in regards to the minimally sufficient grids we have acquired for each QTC variant:

- LineGrid is sufficient for QTC_{B1}, which is very intuitive, considering how the only piece of information we have retained has to do with whether a point approaches or diverges from the other. Still, it is important to now that this is not a one-dimensional grid, because it includes the intersection points between the two circles. This is important, because it allows us to maintain 0 symbols while not remaining static. This is essential for transitions of the form $(0 \rightarrow \pm \rightarrow 0)$
- LineGrid is also sufficient for QTC_{B2}, with the modification that involves different radii for the two circles, according to the velocity property, as described towards the end of this section
- Grid45 is sufficient for QTC_{C1}, which is again quite intuitive. The only information that QTC_{C1} has in addition to that of QTC_{B1} has to do with the side constraints, and Grid45 provides exactly those candidate points
- Finally, ComplexGrid is a grid that also adds points that would satisfy the angular constraints. Thus, when this is combined with variable radii, it becomes sufficient to satisfy all six constraints of QTC_{C2} for time instance τ, as well as their possible continuations

Note that a simple surveying of the target symbols for each case can restrict the grids even further. For exam-

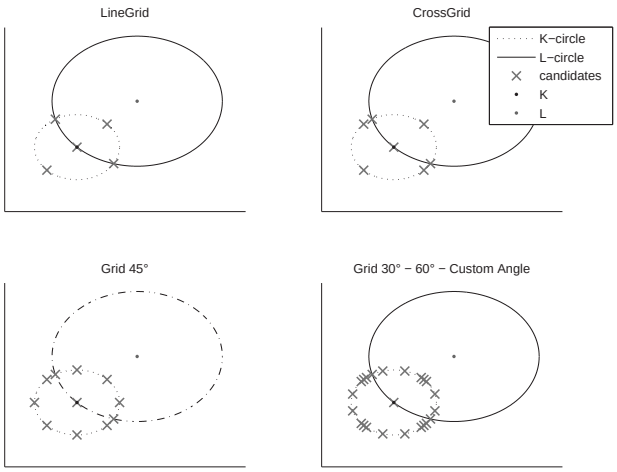


Figure 6: Grids of candidate points for K, according to the QTC variant we are given as input.

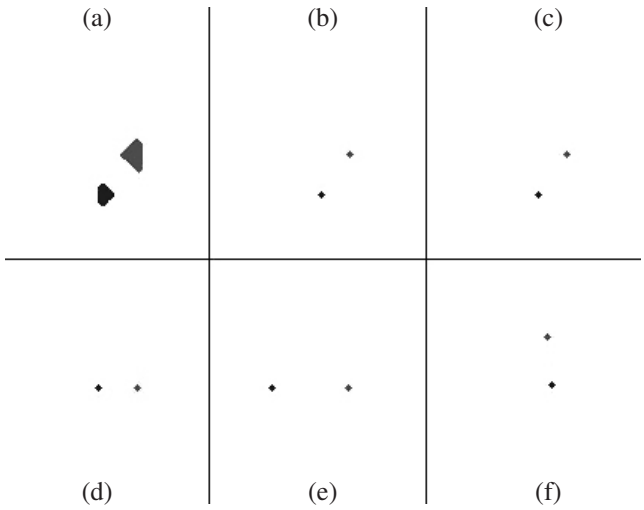


Figure 7: A screenshot from the video of the QTC reconstruction of the Heider and Simmel experiment. *(a)* Segmentation. *(b)* Objects as points. *(c)* 3x interpolation of top middle. *(d)* QTC_{B1} interpolation. *(e)* QTC_{B2} interpolation. *(f)* QTC_{C1} interpolation.

ple, there is no point in returning a candidate that would bring k closer to l if the distance constraint (A) is '+'. This helps speed up the algorithm significantly, especially for the "harder" cases (QTC_{C1} and QTC_{C2}).

One final remaining piece to this puzzle is to select the radius of the circular grid we will be utilizing at each time instant. As a baseline, and since we lie in an arbitrary, custom space, we can always set as the baseline radius $\rho_{base} = 1$. When the input QTC variant includes speed constraint (C) we should however modify ρ_{base} so that it allows for faster or slower movement of the points. For our experiments, this translated to $\rho_{base} = \{0.75, 1.0, 1.25\}$ for the symbols $\{'-', '0', '+'\}$ respectively, when dealing with k. The order will of course have to be inverted for the case of l. However, we have also found that when the two objects happen to drift too far apart, it becomes exceedingly difficult to satisfy all constraints, most notably in the case of the angular constraint F for QTC_{C2}. The solution to this is to further multiply ρ_{base} with a factor resembling the $logsig$ or $1 - logsig$ function, according to whether the distance property is $'-'$ or $'+'$ respectively. This way we manage to maintain the relative difference in radii for the grids of k and l, while also keeping their distances in check.

Examples of QTC Reconstruction

In order to illustrate the usage of the algorithm, we ran the following experiment: The Heider and Simmel video (http://vimeo.com/36847727), which consists of a number of moving triangles and points which are spatially interacting in human-like ways, was imported into MATLAB, segmented, and the resulting trajectories of two of the main characters (triangles) were extracted. Then, the trajectories were analyzed using our QTC encoder. Finally, they were fed into the reconstruction algorithm described in the previous section, and their resulting reconstructed trajectories were visualized. Reconstruction was performed at three levels of qualitative detail: QTC_{B1}, QTC_{B2}, and QTC_{C1}. The results can be seen in our video residing at http://oswinds2.csd.auth.gr/~irini/qtcrecon. A snapshot can be seen in fig. 7. Note that the spatial interactions which take place in the Heider and Simmell video hold quite generally not only for human-human but also for human-robot interaction.

Also, in order to further validate the utility of our QTC reconstruction algorithm for the human-robot case, we have applied it to the real-world human-robots QTC sequences of (Hanheide, Peters, and Bellotto 2012). In particular, we used the QTC sequence from condition 1 of this paper. The reconstruction which successfully captures the qualitative essence of this interaction can be seen in fig. 8.

Future Work

Building, refining, and extending upon the foundation provided here in conjunction with the existing literature, many possible open questions as well as avenues for extension exits. First, although we have investigated the number of possible QTC symbols as a function of threshold settings, the symbol sequence constraints (for pairs, triads, and n-tuples of symbols), are still unknown to us. Second, although we have discussed how to empirically set thresholds that minimize the effect of noise, a neat theoretical model which analytically determines optimal thresholds given a statistical model for the trajectories and the noise (for example, newtonian constant-accelaration trajectories and additive gaussian noise) remains to be derived. Third, the minimality of the grids used in our QTC reconstruction algorithm, could be further investigated. Finally, out of the many possible trajectories that can arise by randomizing the forward- and back-tracking branch selection of our QTC reconstruction algorithm, one would ideally want to chose according to a

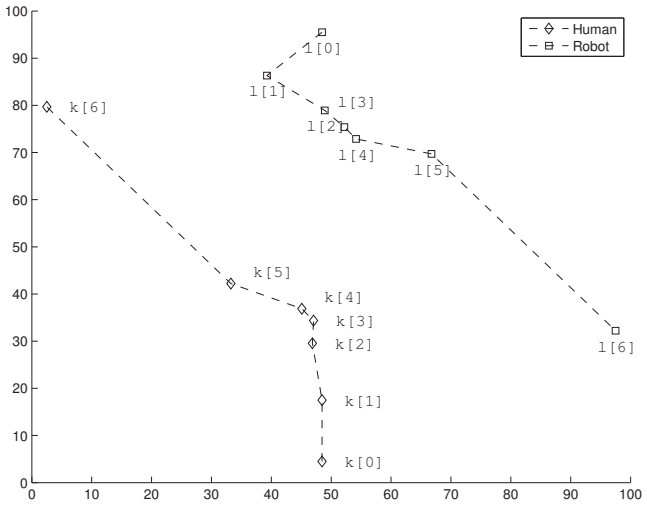


Figure 8: A pair of trajectories that satisfies the given QTC input. Note that we need at least 7 points to get the 5 tuples of symbols for the bidirectional case and that their are drawn in a custom space.

given criterion: for example, suitability for robot motion planning, minimization of total curve length or energy spent, and so on. Most importantly, the QTC reconstruction algorithm could well be extended from pairs of trajectories to triplets and n-tuples of trajectories. All of these questions hold great theoretical interest as well as practical relevance towards the wider application of QTC to human-robot interaction.

Conclusion

In this paper, we have defined and presented an algorithm for solving the inverse problem of qualitative trajectory calculus, namely QTC reconstruction. This algorithm transforms given QTC symbol sequences to trajectories that would have led to them, upon QTC encoding. Also, we have defined and discussed several aspects of real-world noisy discrete-time QTC, as contrasted to the idealized noise-free continuous time QTC that is traditionally covered in the literature. More specifically, we have covered two varieties of discrete-time QTC, namely uni- and bi-directional, and we have discussed abrupt transitions, thresholding, as well as symbolic and syntactical constraints. Our results have shown that a different number of symbols is possible in these cases, as contrasted to the 305 allowable symbols for continuous-time QTC in 2D. Furthermore, we have investigated how the number of symbols and the entropy of the symbol distribution changes as the QTC threshold changes. Finally, after having discussed the importance of these problems for the successful application of QTC to robotics, and having mentioned that existing papers only use hand-crafted behaviors arising out of QTC for robot motion, we have illustrated how our automated QTC reconstruction algorithm can be used not only for generating trajectories for the highly-complex Heider and Simell scenario, but also for the real-world robot trajectories of the (Hanheide, Peters, and Bellotto 2012) paper. Finally, we have discussed the numerous future extensions that are available, to push forward the further widespread application of qualitative trajectory calculus for human-robot interaction.

References

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.

Bellotto, N.; Hanheide, M.; and Van de Weghe, N. 2013. Qualitative design and implementation of human-robot spatial interactions. In *Proc. of Int. Conf. on Social Robotics (ICSR)*.

Bellotto, N. 2012. Robot control based on qualitative representation of human trajectories. In *AAAI Spring Symposium – Designing Intelligent Robots: Reintegrating AI*. TR SS-12-02.

Delafontaine, M.; Van de Weghe, N.; Bogaert, P.; and De Maeyer, P. 2008. Qualitative relations between moving objects in a network changing its topological relations. *Information Sciences* 178(8):1997–2006.

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.

Hanheide, M.; Peters, A.; and Bellotto, N. 2012. Analysis of human-robot spatial behaviour applying a qualitative trajectory calculus. In *Proc. of the IEEE Int. Symposium on Robot and Human Interactive Communication (Ro-Man)*, 689–694.

Heider, F., and Simmel, M. 1944. An experimental study of apparent behavior. *The American Journal of Psychology* 57(2):243–259.

Mather, G., and Murdoch, L. 1994. Gender discrimination in biological motion displays based on dynamic cues. In *Proceedings of the Royal Society of London*, 273–279.

Van de Weghe, N.; Tré, G.; Kuijpers, B.; and Maeyer, P. 2005. The double-cross and the generalization concept as a basis for representing and comparing shapes of polylines. In Meersman, R.; Tari, Z.; and Herrero, P., eds., *OTM Workshops: On the Move to Meaningful Internet Systems 2005*, volume 3762 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 1087–1096.

Van de Weghe, N. 2004. *Representing and Reasoning about Moving Objects: A Qualitative Approach*. Ph.D. Dissertation, Ghent University.

Zhan, B.; Monekosso, D. N.; Remagnino, P.; Velastin, S. A.; and Xu, L.-Q. 2008. Crowd analysis: a survey. *Machine Vision and Applications* 19(5-6):345–357.