



A method based on the Levenshtein distance metric for the comparison of multiple movement patterns described by matrix sequences of different length

Jasper Beernaerts^{a,*}, Ellen Debever^a, Matthieu Lenoir^b, Bernard De Baets^c, Nico Van de Weghe^a

^a Department of Geography, Ghent University, Krijgslaan 281 (S8), Ghent 9000, Belgium

^b Department of Movement and Sports Sciences, Ghent University, Watersportlaan 2, Ghent 9000, Belgium

^c Department of Data Analysis and Mathematical Modelling, Ghent University, Coupure links 653, Ghent 9000, Belgium



ARTICLE INFO

Article history:

Received 4 September 2017

Revised 19 June 2018

Accepted 10 July 2018

Available online 10 August 2018

Keywords:

Moving objects

Qualitative Trajectory Calculus (QTC)

Similarity analysis

Spatiotemporal modelling

ABSTRACT

This paper introduces a new method for calculating distances between matrix sequences of different length. Matrix sequences are often used in spatiotemporal analysis, a field of research combining topological frameworks and temporal reasoning. The Qualitative Trajectory Calculus, a spatiotemporal calculus using matrix sequences, is used as an example in this paper. Since current approaches for comparing sequences are not suitable to calculate distances between matrix sequences of different length, a new method is proposed. The method is based on sequence alignment and uses the Levenshtein distance metric as starting point. The method is presented in general at first, followed by an application to QTC-matrix sequences and a three cushion billiards movement pattern detection example. Possible applications in expert and intelligent systems are discussed, such as human-robot interaction with multiple robots, and automated traffic analysis of (autonomous) vehicle interactions. The proposed method, however, can be applied in a broad range of domains, whenever distances need to be calculated between matrix sequences of different length.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

The Qualitative Trajectory Calculus (QTC) introduced by Van de Weghe (2004) is frequently used to describe the movement behaviour of Moving Point Objects (MPOs). It can be used, just as other qualitative calculi such as RCC (Cohn, Bennett, Gogday, & Gotts, 1997) and REMO (Relative Motion) (Laube, Van Kreveld, & Imfeld, 2004), to describe and compare the movement of MPOs in a qualitative way. The problem is that, when comparing similar movement events, such as different overtaking manoeuvres in traffic (Van de Weghe, Cohn, De Maeyer, & Witlox, 2005), these events often have a slightly different temporal duration. Consequently, when working equal-interval based, the qualitative representations of the movement events might differ in length, making their comparison difficult. While switching to an event-based approach enables the comparison of events by alignment of their

qualitative representations of equal length, it also limits the ability of the (qualitative) calculus to register subtle differences between sets of moving points during movement intervals. In order to compare qualitative representations of sets of MPOs with different lengths, a distance calculation method for matrix sequences based on the Levenshtein distance metric (Levenshtein, 1966) is introduced in this paper. This method calculates distances between sequences by counting the number of edit operations needed to align both sequences, thereby allowing differences in length between the sequences. Whereas existing sequence alignment methods (SAMs) use the Levenshtein distance metric to align sequences with atomic (i.e. non-divisible) elements, the proposed method allows to align sequences that are composed of matrices as elements.

By making use of the Levenshtein distance metric, movement patterns with slightly different temporal lengths can be compared when they are represented by a qualitative calculus in the form of matrix sequences. Important to mention is that although this paper will focus on QTC to describe and compare the movement patterns, the proposed method can be applied to a wide range of qualitative representations of, for example, movements. The novel method can

* Corresponding author.

E-mail addresses: Jasper.Beernaerts@UGent.be (J. Beernaerts), ellendebever@hotmail.com (E. Debever), Matthieu.Lenoir@UGent.be (M. Lenoir), Bernard.DeBaets@UGent.be (B. De Baets), Nico.VandeWeghe@UGent.be (N. Van de Weghe).

be applied to any pair of matrix sequences if a distance metric for transitions between the different qualitative values of the qualitative calculus used in those matrices, such as a conceptual distance table, is provided.

2. Background

2.1. Spatiotemporal reasoning

The research presented in this paper originates from the field of spatiotemporal reasoning about moving objects, a field where matrices are often (Cohn et al., 1997; Van de Weghe, Cohn, De Tré, & De Maeyer, 2006) used to store information on relations between objects. If the temporal evolution of these relations is considered, these matrices can be placed into a sequence, taking into account the changes over time. The research presented in this paper originates from the lack of an adequate method for calculating the distance between sequences of matrices of different length.

Looking at temporal reasoning, an important work is the overview of Allen (1981) for reasoning with temporal knowledge, based on an interval-based representation. This influential work was further enhanced by Galton (1990), making it suitable for assessing continuous changing temporal phenomena. For purpose, Galton introduced a series of modifications to Allen's work, including the broadening of the original temporal ontology to include time instants at the same level as the time intervals (Galton, 1990).

During the late 20th century as well as in more recent years, a lot of research has been done in the field of spatial reasoning. Qualitative spatial reasoning in particular has seen a huge growth (Cohn, 2005), thereby enabling detailed analyses on big datasets with limited computation power. One of the reference works for describing qualitative relations between objects is the topological relations framework of Egenhofer (1989). As it was later extended to include geographical regions (Egenhofer & Herring, 1991), relations between fuzzy objects (Dijkmeijer & de Hoop, 1996), three-dimensional relations (Billett & Zlatanova, 2003) and region-region relations (Egenhofer, 2010), it is an excellent example of the emerging need for qualitative spatial reasoning. Research on topological relations such as that of Egenhofer and Shariff (1998) was taken further by Bera and Claramunt (2003), with the introduction of topology-based measures that describe nearness relationships between regions in a spatial system. A similar framework of qualitative spatial reasoning, based on the logic of regions, is the Region Connection Calculus (RCC), incorporating complex concepts such as regions with uncertain boundaries (Randell, Cui, & Cohn, 1992). RCC, which describes relations between regions, consists of eight possible relations in its most common form (RCC8). Worboys (1994), however, stated that spatial relations often have temporal relationships, and that rather than posed in a subordinate role, this temporal information should be incorporated in spatial models, as a unified model of spatiotemporal relations.

The interest in qualitative spatiotemporal reasoning about moving objects has seen a huge growth in the last decade, with considerable efforts in the formalisation of motion. Examples of this can be found in Ibrahim and Tawfik (2007), Muller (2002), and Kurata and Egenhofer (2009). In the following paragraph, a qualitative framework for reasoning about moving objects, called the Qualitative Trajectory Calculus (Van de Weghe et al., 2005), will be discussed in more detail since it will be used in an example of our proposed method.

2.2. Qualitative trajectory calculus

The Qualitative Trajectory Calculus (QTC), introduced by Van de Weghe in 2004, describes the relative motion of disconnected moving objects, thus providing an answer to many

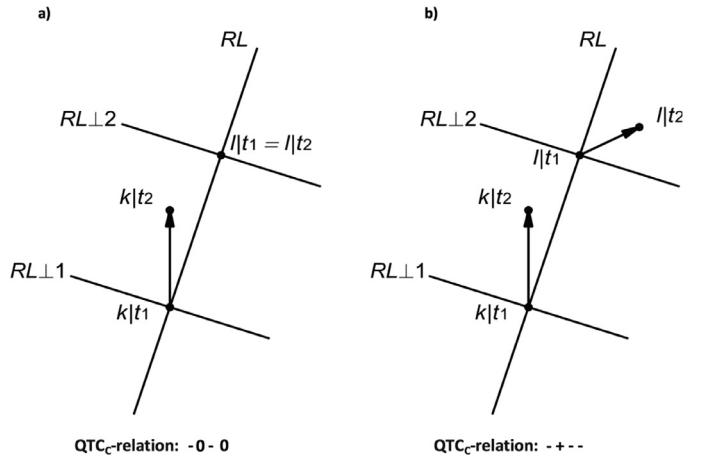


Fig. 1. Double-cross variant of QTC (Van de Weghe et al., 2005).

trajectory-related questions. The theoretical framework of QTC has been well studied and documented by composition tables (Van de Weghe et al., 2006) and conceptual neighbourhood diagrams (Van de Weghe & De Maeyer, 2005). Several variants of QTC have been described (Van de Weghe et al., 2005). In this paper, the QTC_C -variant is used as an example of a qualitative calculus for describing movement of MPOs, because of its practicability and precision of describing relative trajectories of moving objects (Delafontaine, Cohn, & Van de Weghe, 2011).

2.2.1. QTC double-cross

The QTC double-cross variant (Van de Weghe et al., 2005), which we will denote as QTC_C in the remainder of the paper, uses in its most basic version four characters to describe the relative movement between two MPOs. It is an extension of QTC basic (QTC_B , Van de Weghe et al., 2005) including two extra characters used for describing relative angles of the movement direction, enabling the calculus to describe the movement path of MPOs in more detail. In cases with a limited number of MPOs, such as the billiards example in this paper, QTC_C is more suitable than the more basic QTC_B (Delafontaine et al., 2011).

The QTC_C -relation of two MPOs, *k* and *l*, during a certain time interval $T = [t_1, t_2]$ is a tuple (a, b, c, d) consisting of qualitative symbols that share the threefold domain $D = \{-, 0, +\}$. Character *a* is the movement of *k* with respect to *l* during interval T with respect to the first perpendicular reference line ($RL \perp 1$), whereas *b* is the movement of *l* with respect to *k* during the same interval T with respect to the second perpendicular reference line ($RL \perp 2$). The reference lines are part of the Double-Cross concept (Fig. 1), with $RL \perp 1$ perpendicular to the connecting line between *k* and *l* at t_1 going through *k*, and $RL \perp 2$ perpendicular to the same connecting line going through *l*. The movement of *k* with respect to *l* during interval T (and vice versa) is defined as (distance constraints):

- '-' if *k* moves towards *l*
- '+' if *k* moves away from *l*
- '0' if *k* is stable in relation to *l* (all other cases)

Character *c* is the movement of *k* with respect to *l* during interval T with respect to the reference line (RL) between both MPOs, whereas *d* is the movement of *l* with respect to *k* during the same interval T with respect to the reference line (RL) between both MPOs (Fig. 1). The movement of *k* with respect to *l* during interval T (and vice versa) is defined as (side constraints):

- '-' if *k* is moving to the left side of RL
- '+' if *k* is moving to the right side of RL
- '0' if *k* is moving along RL

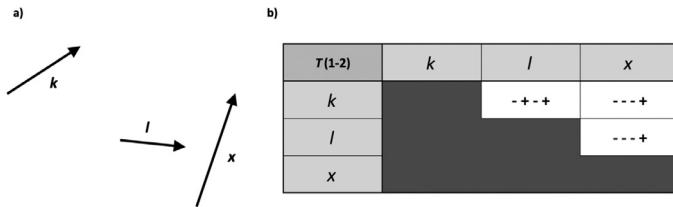


Fig. 2. Describing multiple MPOs with QTC_C by means of a QTC -matrix.

To determine whether the object is moving to the left or right side of the reference line, it is important to know that the observer is located at k and is directed towards l (and vice versa for object l). The combination of three possibilities ('-', '+' and '0') for four characters results in 81 (3^4) possible QTC_C -relations (Van de Weghe et al., 2005).

2.2.2. Multiple MPOs and timestamps

As described above, QTC_C only takes two MPOs into account, during a specific time interval. When considering the movement of multiple MPOs (Fig. 2a), a matrix can be used to collect all pairwise comparisons between the different objects (Fig. 2b). An element (i,j) of this matrix represents the QTC_C -relation between MPOs i and j (Van de Weghe et al., 2005). The movement interaction between vectors k and l during the time interval $T(1-2)$ can be described by a '-++-'-relation in QTC_C . The first character (-) indicates that vector k is moving towards the starting point of vector l , the second character (+) indicates that vector l is moving away from the starting point of vector k . The third and fourth characters indicate the directions of the vectors compared to the reference line connecting the starting points of both vectors, as described in the previous subsection.

When the movement of multiple objects is described over time, a sequence of matrices can be used to store the QTC_C -relations for all successive time intervals. Real-world movement interactions can thus be described by means of matrix sequences, where the number of matrices in the sequence is dependent on the temporal duration of the event (as can be seen in Fig. 6). This is why, when comparing a series of similar events such as the overtaking example (Van de Weghe et al., 2005), small differences occur in the length of the QTC_C -representations. These can be caused by differences inherent to the movement patterns of that event, but also because identical trajectories were performed at different speeds when they are described in an equal-interval based manner.

2.3. Distance calculation

For comparing different movement events, a distance needs to be calculated between the qualitative representations of these events. Note that this distance is not the spatial distance but rather a measure of dissimilarity between the qualitative representations. After obtaining these distances, a similarity analysis can be performed (Gentle, 2007). This similarity analysis usually results in a grouping of the most similar events and can be elucidated by means of visual representations such as a dendrogram in hierarchical clustering. In this section, we will present some of the most common (basic) techniques used for distance calculation between sequences. First of all, we will give an introduction to basic alignment methods. These assume that the elements of the sequences are fixed, meaning that they are elementary building blocks of the sequence and cannot be further divided, and are finite in number. A simple and common example of such fixed building blocks of sequences are the letters of a word. In the remainder of this paper, we will refer to these undividable building blocks of sequences as 'sequence elements'. In case of the basic alignment methods, pairwise distance calculation can be performed based on a conceptual

B	I	L	L	I	A	R	D
B	I	L	-	J	A	R	T

✓ ✓ ✓ - ✗ ✓ ✓ ✗

Fig. 3. Pairwise alignment of the English word 'BILLIARD' and its Dutch translation 'BILJART' (after Delafontaine, Versichele, Neutens, & Van de Weghe, 2012).

distance table. Secondly, an introduction to the distance calculation between matrix sequences is provided. Where the former methods can calculate distances between sequences of different length, current approaches for matrix sequences do not have this ability. However, when describing similar events by means of a qualitative calculus such as QTC_C , the qualitative representation, in this case sequences of matrices, do not necessarily have the same length, even if the user would classify them as similar based on a visual analysis of the movements.

2.3.1. Basic sequence alignment methods

Sequence Alignment Methods (SAMs) are methods that equate two or more sequences of elements by applying a set of qualified operations (Morrison, 2010). Such methods try to achieve optimal alignments by employing dynamic programming algorithms to minimise a distance metric (Wilson, 2008). There are two types of sequence alignment algorithms: global alignment methods force alignment to cover the entire length of the sequences, while local alignment methods base alignment on similar parts in the sequences, even if the entire sequences differ significantly (Schlich, 2003). Important to notice is that basic sequence alignment methods enable alignment of sequences of different length, opposed to so-called lock-step measures (Ranacher & Tzavella, 2017). The need for SAMs for calculating distances between sequences arose in biology, where sequences of DNA (or RNA, proteins, etc.) strings were subject of similarity analysis (Rosenberg, 2009). A breakthrough in the computation of sequence alignment occurred in 1966 when Levenshtein introduced a distance metric based on edit operations (Levenshtein, 1966). Numerous new methods for sequence alignment were proposed over time, taking the Levenshtein method as a starting point. As the novel method proposed in this paper also takes the Levenshtein method as base, a thorough overview of the technique is provided here. A good understanding of the Levenshtein method is therefore crucial for the comprehension of the remainder of the paper and the given examples. Levenshtein calculates a distance based on edit operations, commonly called the *edit distance*, and can be defined as the minimum cost of changing one string into another by applying a sequence of weighted edit operations usually including *identity*, *substitution*, *insertion* and *deletion*. As the latter two always occur together, they are usually called *indels*.

The suitable operations have specific costs that are chosen by the user. In terms of distance, the identity operations typically have no cost, while substitutions and *indels* are usually associated with positive costs. To display the editing process, the gap representation (Fig. 3) is often used. This representation places strings one above the other, with a gap in the second word for every insertion corresponding with a deletion in the first word (Wilson, 2008). As can be seen in Fig. 3, a minimum of three steps is needed to go from [BILLIARD] to [BILJART].

For longer examples, it is considerably more difficult to define the minimum number of steps and hence algorithms are needed to calculate the distance. One of the most popular algorithms for computing the edit distance between two strings, based on dynamic programming, is the algorithm of Needleman and Wunsch (1970). Despite the vast number of algorithms for edit distance calculation, it can be seen as a generally accepted standard (Tönges, Perrey, Stoye, & Andreas, 1996; Waterman, 1984).

		SOURCE								
		B	I	L	L	I	A	R	D	
TARGET	0	2	4	6	8	10	12	14	16	
	B	2	0	2	4	6	8	10	12	14
	I	4	2	0	2	4	6	8	10	12
	L	6	4	2	0	2	4	6	8	10
	J	8	6	4	2	1	3	5	7	9
	A	10	8	6	4	3	2	3	5	7
	R	12	10	8	6	5	4	3	3	5
	T	14	12	10	8	7	6	5	4	4

Fig. 4. Computation of the minimum edit distance for the strings [BILLIARD] and [BILJART] and one possible alignment (yellow).

Pairwise alignment is basically the process of equating a source and a target string using a set of accepted operations (Wilson, 2008). Following the Needleman & Wunsch method, the first step is to write the elements of the so-called source and target sequence in the margins of a comparison table of $n+1$ rows and $m+1$ columns (Fig. 4), with n and m respectively equal to the number of characters in the source and target sequence. A cumulative cost is calculated beginning at the upper left cell (0,0), proceeding to the lower right cell ($m+1, n+1$). A diagonal movement represents either an identity or substitution operation. A horizontal movement in the table represents an insertion in the source string or a deletion in the target string. A vertical movement in the table represents a deletion in the source string or an insertion in the target string. The minimum edit distance for the alignment of the two strings can be found in the lower right cell ($m+1, n+1$) of the comparison table.

Having an m -character source string $A[1\dots m]$ and an n -character target string $B[1\dots n]$, every element $E(i,j)$ in the comparison table is the edit distance between the prefixes of length i and j which are $A[1\dots i]$ and $B[1\dots j]$. The four possibilities (identity, insertion, deletion and substitution) are evaluated and the minimum value among the four operations is chosen. Note that the substitution cost is zero in the case of identical characters and the cost for insertion is equal to the cost of a deletion (written as *indel*).

$$E(0, 0) = 0, E(i, j) = \min \begin{cases} E(i - 1, j - 1) + \text{substitution cost} \\ E(i, j - 1) + \text{indel cost} \\ E(i - 1, j) + \text{indel cost} \end{cases}$$

Starting from the lower right cell, every non-decreasing path towards the upper left cell in the comparison table represents an alignment of the two sequences, whereby the best alignment is composed of optimal subalignments. There is always at least one valid path available, yet in most cases there will be several (Needleman & Wunsch, 1970).

Fig. 4 demonstrates the computation of the minimum edit distance for the strings [BILLIARD] and [BILJART] with one possible alignment, visualised by the yellow path. When applying a constant indel cost of two and a constant substitution cost of one, for the ease of understanding, the minimum edit distance to transform [BILLIARD] into its Dutch equivalent (and vice versa) equals 4. Note

	Billiard [EN]	Biliardo [IT]	Biljart [NL]	Bilardo [TR]
Billiard [EN]	0	4	4	6
Biliardo [IT]	4	0	4	2
Biljart [NL]	4	4	0	4
Bilardo [TR]	6	2	4	0

Fig. 5. Distance matrix containing the minimum edit distance between pairs formed by the strings [Billiard], [Biliardo], [Biljart] and [Bilardo].

that this alignment consists of the same series of actions (insertion, deletion and substitution) as the alignment presented in Fig. 3.

When three or more sequences need to be aligned, a series of pairwise alignments is performed. This method is often referred to as progressive alignment (Wilson, 2006) and results in a series of pairwise distance values between the different sequences, which can be stored in the form of a distance matrix. Fig. 5 contains a distance matrix representing the minimum edit distances between different translations of the word [Billiard] in respectively English (EN), Italian (IT), Dutch (NL) and Turkish (TR). They were calculated according to the Needleman & Wunsch method, with a constant *indel* cost of two and a constant substitution cost of one, similar to the example presented in Fig. 4.

Ever since the introduction of the edit distance, multiple variants were introduced, such as the Edit Distance on Real Sequences (EDR) which calculates the edit distance based on the Euclidean distance (Chen, Özsü, & Oria, 2005) and, more specifically for the comparison of trajectories, the Normalized Weighted Edit Distance (NWED; Dodge, Laube, & Weibel, 2012), which takes speed and acceleration of objects into account. In a similar endeavour to find the optimal alignment for sequences that represent movements differing in speed, Dynamic Time Warping (DTW), was introduced by Keogh and Pazzani (2000).

2.3.2. Sequence alignment for matrix sequences

In the previous section, an introduction to basic sequence alignment methods was given.

Often these methods assume letters or other non-divisible elements as building blocks of the sequences. In biology, DNA sequences are represented by sequences of letters and alignment between them is thus done on sequences of letters. In other fields, such as sociology, letters are often used to represent sociological phenomena. Sequence alignment as such was first implemented in sociology by Abbott (1995), in the analysis of career paths. This triggered a general interest in SAM techniques by sociologists (Abbot & Tsay, 2000), where sequence alignment was seen as a promising methodology for analysing the sequential aspect of human activities in space and time (Delafontaine, Chavoshi, Cohn, & Van de Weghe, 2012). On the crossroads of sociology and geography, for example, Shoval and Isaacson (2007) used the Levenshtein distance metric for sequence alignment of human activity, while Kwan, Xiao, and Ding (2014) extended this method to include multiple dimensions of human activity. Stehle and Peuquet (2015) extended the basic sequence alignment to align partial patterns using temporal intervals and fuzzy matching, exemplified by the alignment of political transitions of different states during the Arab Spring. For some phenomena, such as the QTC

TRAJECTORIES		QTC REPRESENTATION				TEXT REPRESENTATION																																			
VARIANT 1		<table border="1"> <thead> <tr> <th>I(1-Z)</th><th>B</th><th>N</th><th>X</th></tr> </thead> <tbody> <tr> <td>B</td><td></td><td>----</td><td>----</td></tr> <tr> <td>N</td><td></td><td>++-</td><td>++-</td></tr> <tr> <td>X</td><td></td><td>+++</td><td></td></tr> </tbody> </table>				I(1-Z)	B	N	X	B		----	----	N		++-	++-	X		+++		<table border="1"> <thead> <tr> <th>I(2-3)</th><th>B</th><th>N</th><th>X</th></tr> </thead> <tbody> <tr> <td>B</td><td></td><td>+--</td><td>++-</td></tr> <tr> <td>N</td><td></td><td></td><td>----</td></tr> <tr> <td>X</td><td></td><td></td><td></td></tr> </tbody> </table>				I(2-3)	B	N	X	B		+--	++-	N			----	X			
I(1-Z)	B	N	X																																						
B		----	----																																						
N		++-	++-																																						
X		+++																																							
I(2-3)	B	N	X																																						
B		+--	++-																																						
N			----																																						
X																																									
VARIANT 2		<table border="1"> <thead> <tr> <th>I(1-Z)</th><th>B</th><th>N</th><th>X</th></tr> </thead> <tbody> <tr> <td>B</td><td></td><td>++-</td><td>---</td></tr> <tr> <td>N</td><td></td><td></td><td>---</td></tr> <tr> <td>X</td><td></td><td></td><td>++</td></tr> </tbody> </table>				I(1-Z)	B	N	X	B		++-	---	N			---	X			++	<table border="1"> <thead> <tr> <th>I(1-Z)</th><th>B</th><th>N</th><th>X</th></tr> </thead> <tbody> <tr> <td>B</td><td></td><td>H</td><td>I</td></tr> <tr> <td>N</td><td></td><td></td><td>I</td></tr> <tr> <td>X</td><td></td><td></td><td></td></tr> </tbody> </table>				I(1-Z)	B	N	X	B		H	I	N			I	X			
I(1-Z)	B	N	X																																						
B		++-	---																																						
N			---																																						
X			++																																						
I(1-Z)	B	N	X																																						
B		H	I																																						
N			I																																						
X																																									
						<table border="1"> <thead> <tr> <th>Alignment</th><th>D</th><th>D</th><th>E</th><th>F</th><th>G</th><th>D</th></tr> <tr> <th></th><th>H</th><th>I</th><th>I</th><th></th><th></th><th></th></tr> </thead> </table>				Alignment	D	D	E	F	G	D		H	I	I																					
Alignment	D	D	E	F	G	D																																			
	H	I	I																																						

Fig. 6. Letter representation and alignment of two QTC_C-matrix sequences.

representation of multiple MPOs as presented in Section 2.2.2, it is less straightforward to use a character sequence representation. In comparison with existing methods to compare spatio-temporal trajectories of single MPOs based on sequence alignment techniques (Dodge et al., 2012), the comparison of different QTC-matrix sequences can be seen as the alignment of two or more trajectories of multiple MPOs. In this section, we will present methods and techniques that are available to apply SAM on matrix sequences, thus providing distance values for a similarity analysis of a number of matrix sequences.

The most straightforward solution for applying SAM to matrix sequences is by replacing the whole matrix by another qualitative representation such as a letter. In that case, a conceptual distance table, containing all distances between all the different (possible) representations would be needed, which might pose a problem when working with huge matrices containing a lot of elements, necessitating huge calculation power. In the rather basic but common case of analysing movement relations between six objects in QTC_C, there are 81¹⁵ unique QTC-matrices. A bigger problem is that the matrices would be seen as a whole, i.e. as elementary building blocks of the sequences, and thus no sequence alignment would be applied on the inner elements in the matrix.

A different solution is to transform the matrix representations to text or other qualitative representations by taking every single element of the matrix as a base. In other words, the sequences would be stripped of their matrix representation, allowing to form sequences containing a series of singular elements which were once the elements that could be found in the cells of the different matrices. When applying this method, basic sequence alignments methods could be applied to the sequences. Depending on the number of matrices in each sequence, the letter representation would differ in length, yet this would not pose a problem for basic SAMs as argued in Section 2.2.1. An example of this method is given in Fig. 6, where two movements of MPOs are shown. On the right of these visualisations, QTC_C-matrix representations for both movements are shown, containing a different number of matrices due to the difference in temporal duration. The matrix representations are being transformed by replacing all matrix elements by letters (text representation). The actual alignment according to this method is shown at the bottom of Fig. 6, where the two letter representations are being aligned, which can be done by the classical SAM methods mentioned earlier.

There is, however, a downside to this approach, because ignoring the matrix representation may cause some of the valuable information to get lost. The fact that information is stored in matrices means that there is more semantic information included. Corresponding matrices store information about the same timestamps, such as matrices I(1-2) of variant 1 and I(1-2) of variant 2 in Fig. 6. Corresponding cells in different matrices store relations between the same objects, such as the cells containing the QTC_C-relation between object B and X in matrices I(1-2) in variant 1 and I(1-2) in variant 2. If the different matrix sequences have identical lengths, an alternative for SAM is the pairwise comparison of the elements in the matrix cells. As such, a distance value can be calculated if a conceptual distance table for the possible elements in the matrix is available.

We can conclude that there is currently no optimal sequence alignment method for matrix sequences of different length.

3. Methodology

In this section, a novel method for distance calculation between matrix sequences, based on matrix sequence alignment, will be presented. First, this will be done for the general case of matrix sequences, regardless of the qualitative calculus used in the matrices. Later, the proposed methodology will be applied more in detail to QTC-matrix sequences of different length. The discussion section will present additional qualitative calculi that could be used with this methodology.

3.1. Levenshtein distance calculation for matrix sequences

As mentioned before, basic sequence alignment methods need to be enhanced in order to compare matrix sequences of different length. New methods should thereby take into account special features of matrices, such as dependence between cells from different matrices with similar column and row index and temporal relations between matrices located at the same position in different matrix sequences. The proposed method takes the Levenshtein (1966) and the Needleman and Wunch (1970) methods as starting point for the distance calculation. Compared to the basic Levenshtein approach as presented earlier, there are some major differences in the novel methodology.

In the basic sequence alignment methods, two words can be compared and aligned by substituting, deleting and inserting letters. In that case, the words are the sequences and the letters

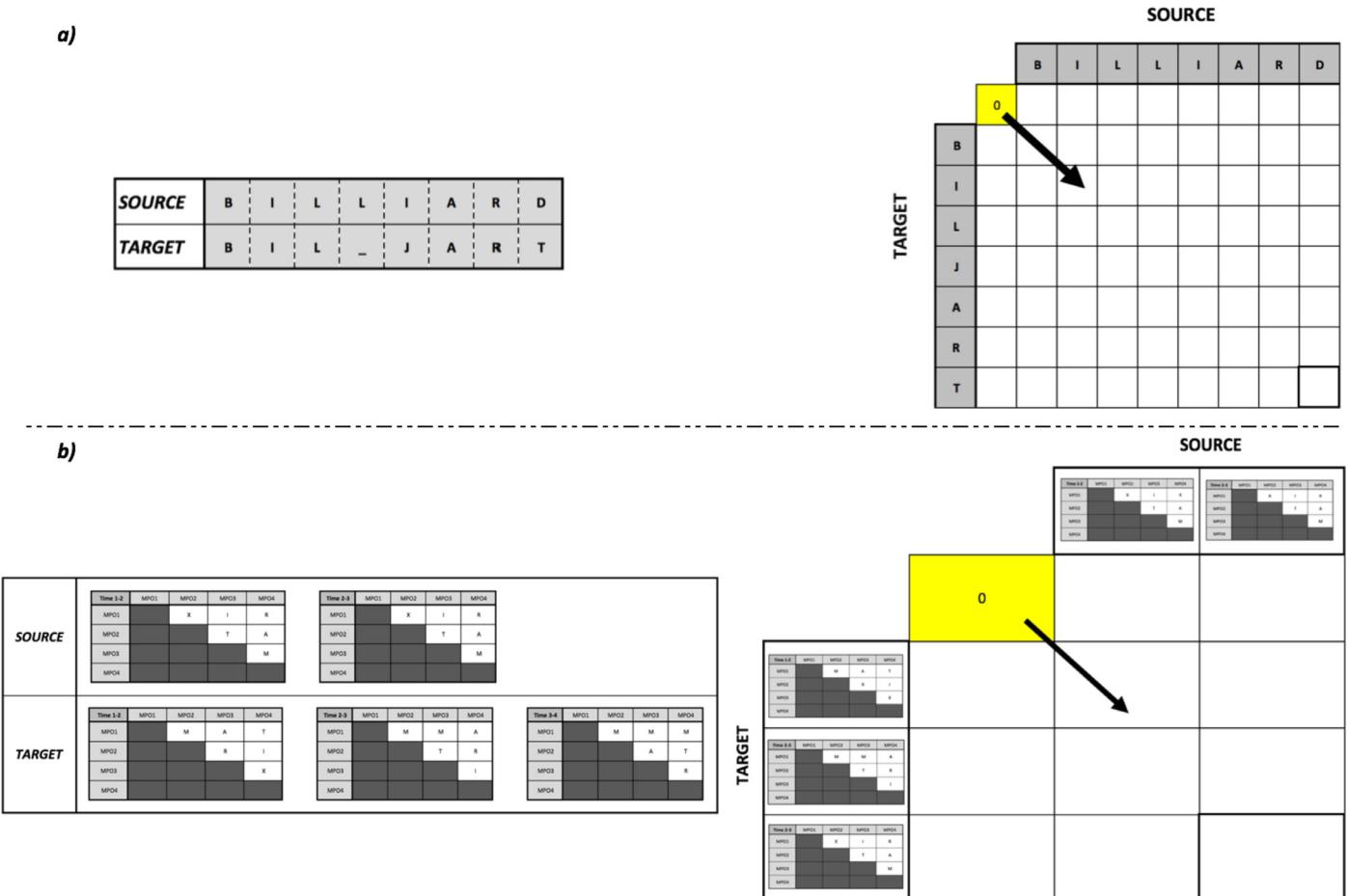


Fig. 7. Basic alignment of two strings (a) and novel methodology for alignment of two matrix sequences of different length (b).

can be seen as the sequence elements. In the novel approach, matrix sequences are aligned, where the full matrix sequences can be seen as the sequences and the individual matrices can be seen as sequence elements (Fig. 7). By this approach, the non-divisible character of the sequence elements, as described in Section 2.2, is being violated in our proposed methodology. For the ease of comparison though, we will further use the term ‘sequence elements’ in the remainder of this paper for both cases.

If, when comparing two matrices, all cells of the matrices containing the characters of the calculus, are identical, the substitution cost is zero. If some of the cells are not identical, the distance between both matrices can be calculated, in the most basic version of the method based on pairwise comparison of the characters. Dependent on the calculus used to describe movement between the MPOs, this can be done ‘on-the-fly’ by calculating the distance between every pair of non-identical characters upon occurrence or all the possible transitions between characters can be calculated in advance and saved in a conceptual distance table. It depends on the number of possible transitions (combination of different characters) whether it is more feasible to calculate this on-the-fly rather than looking it up from a pre-computed conceptual distance table. The indel cost of a whole matrix is equal to the maximum possible substitution cost between two matrices.

Our novel method imposes a restriction on the matrix dimensions, demanding that both matrix sequences should contain matrices having the same dimensions. Because of that, it is possible to perform the rather basic approach of pairwise comparison, since each matrix contains the same number of elements. To fine-tune the distance calculation, matrices can be compared by applying the

Levenshtein method on the letter representation of their elements. Based on the preferences of the user, by applying different values for the indel and substitution costs, countless variants of distance calculation are possible. One could argue that by applying an infinite indel cost on the internal Levenshtein comparison of the matrices, thereby disabling the deletion and insertion options in the alignment, the method would reduce to the pairwise comparison mentioned earlier. Thus the pairwise comparison of two matrices can be seen as a Levenshtein-based comparison with an infinite indel cost.

Where the basic Levenshtein algorithm (see Section 2.3.1) enables to align and provide a distance value for words with a different number of letters, the novel method can provide an alignment for matrix sequences with a different number of matrices, thereby incorporating the special relations (temporal, geographical) that are linked to this matrix representation. Weights for differences between the characters that compose the matrices can be chosen freely for every calculus and can be changed for different applications of every calculus. The weights can be derived from a conceptual distance table.

3.2. QTC-matrix sequences

As presented earlier, QTC-matrices can be used to store the qualitative representations of multiple MPOs through time. A sequence alignment method for this kind of data enables distance calculation between different intervals of moving objects and thus measure the similarity between them. QTC in combination with SAM has already been used to analyse similar-

QTC Character	-	0	+
-	0	1	2
0	1	0	1
+	2	1	0

Fig. 8. Conceptual distance matrix for QTC-characters.

ity between different human-robot interactions (Hanheide, Peters, & Bellotto, 2012) and movement patterns of body parts of samba dancers (Chavoshi, De Baets, Neutens, De Tre, & Van de Weghe, 2015). An important difference, however, is that these approaches cannot deal with interactions between multiple MPOs since the QTC-representations are analysed in specific pairs represented by separate QTC-sequences. As such, the methods proposed by Chavoshi and Hanheide can be seen as a special (more limited) case of the more general method that is presented in this paper. This method is much needed, because often relations between more than two MPOs are described by means of QTC-matrices (Delafontaine et al., 2012). Moreover, QTC-matrices tend to describe movements of MPOs more precisely, since for one MPO multiple QTC descriptions with respect to other objects are given. The combination of these constraints on the movement of that MPO results in a more precise description of its movement.

Two QTC-matrix sequences can be aligned as shown in Fig. 7(b). Each cell of these matrices contains the QTC-relation between the corresponding objects indicated in the row and column header (Fig. 2). Aligning QTC-matrices requires a conceptual distance matrix for scoring the distance between different QTC-characters, displayed in Fig. 8. In this case, the conceptual distance is a similarity measure for two QTC-relations and is calculated by counting the number of changes in the symbols of the QTC-representations (Van de Weghe & De Maeyer, 2005). When two QTC-characters are identical, the conceptual distance is zero, the conceptual distance between '0' and '-' or '+' is one. The conceptual distance between '+' and '-' is two because direct transition from '+' to '-' or vice versa is impossible due to continuity of the movement (Galton, 2001). The total conceptual distance between two QTC-relations can be calculated by summing up the conceptual distances over all relation characters (Chavoshi et al., 2015). When using qualitative calculi, summing up the conceptual distances will always result in a finite overall distance value. This is important because the indel cost was defined as the maximum possible substitution cost for the whole matrix (see Section 3.1). Using a quantitative calculus could result in an infinite indel cost, impeding the use of indels when aligning two matrix sequences, thereby not guaranteeing optimal alignment.

The Levenshtein distance calculation is applied to the matrix sequences (Fig. 9(a)), including conceptual distance calculations between each pair of individual matrices. These conceptual distance calculations, with one of the calculations shown in Fig. 9b, provide the distances used for calculating the minimum edit distance for the alignment of the two matrix sequences (Fig. 9(a)). Note that by using the conceptual distance calculation presented in Fig. 9(b), a one-on-one identity of the QTC_C-characters of the different matrices is needed for a distance of zero, taking into account the underlying temporal and geographical dependencies of cells of corresponding matrices or cells with identical row and column indices.

4. Example: three cushion billiards

In this example, the use of the Levenshtein distance calculation for matrix sequences will be demonstrated by performing a spatiotemporal comparison of a series of three cushion billiards shots.

4.1. Three cushion billiards

Three cushion billiards is a variant of the more popular carom billiards. The game is played with three balls, at start positioned as displayed in Fig. 10. The goal of the game is for the player to play the cue (white) ball and to hit the two object balls (red and yellow), while hitting three of the sides with the cue ball before touching the last object ball. According to the success of the shot, points are given to the player. After the first shot, the game continues with the three balls positioned as they were at the end of the first shot (Cohen, 2002).

4.2. Dataset

In this example only the opening shot of the game is considered. A series of 55 shots were generated by an online tool (<http://www.casualarena.com/games-french.php>), applying the most popular opening shot technique used by casual players (Byrne, 1998). Coordinates (x,y) of the centroids of the three balls were logged at a temporal resolution of 20Hz. The 55 shots got named by the number of the shot (1–55), followed by a letter (S, R or L) according to the type of the shot (see Figs. 11 and 12). Fourteen of the 55 opening shots are from the type 'success' (label 'S'), meaning that the goal of hitting the two object balls and three sides in between, was achieved. The other 41 shots are divided in two types, according to their characteristics and their trajectories, as shown on the left of Fig. 12. The 'pass on the right'-type (label 'R') contains shots where the white ball hit the red ball but passed on the right of the yellow ball. It contains a subgroup called 'yellow touched' of shots with similar properties, with the difference that the red ball hit the yellow one. Shots of the 'pass on the left'-type (label 'L') consist of the white ball first hitting the red one and thereafter passing on the left of the yellow ball.

4.3. Experiment

For performing a spatiotemporal analysis on the dataset, each opening shot can be described with the Qualitative Trajectory Calculus, by means of a sequence of QTC_C-matrices. The QTC_C-representations can then be compared in order to analyse similarities between different shots, which might give an insight in the requirements to perform a successful opening shot or even to facilitate automated recognition of shot types.

Because of differences in how hard the cue ball is played, small variations in where the cue ball hits the edge of the field, or even different playing styles including the addition of 'effect' to the cue ball, shots of the same type do not always take the same amount of time. When applying a constant temporal resolution of 20Hz, the QTC_C-representation of the different shots sometimes thus consists of a slightly different number of QTC_C-matrices (fourth column of Fig. 11). Since traditional methods do not allow for the calculation of distances between matrix sequences of different length, this is a perfect case to test the novel distance calculation method.

One might propose a transformation from the equal-interval-based dataset to an event-based logging of the coordinates to cope with the differences in temporal length. However, only logging coordinates at specific points, e.g. when one of the balls hits one of the sides of the field, does not always result in shots with identical lengths, as different numbers of those events might occur in the

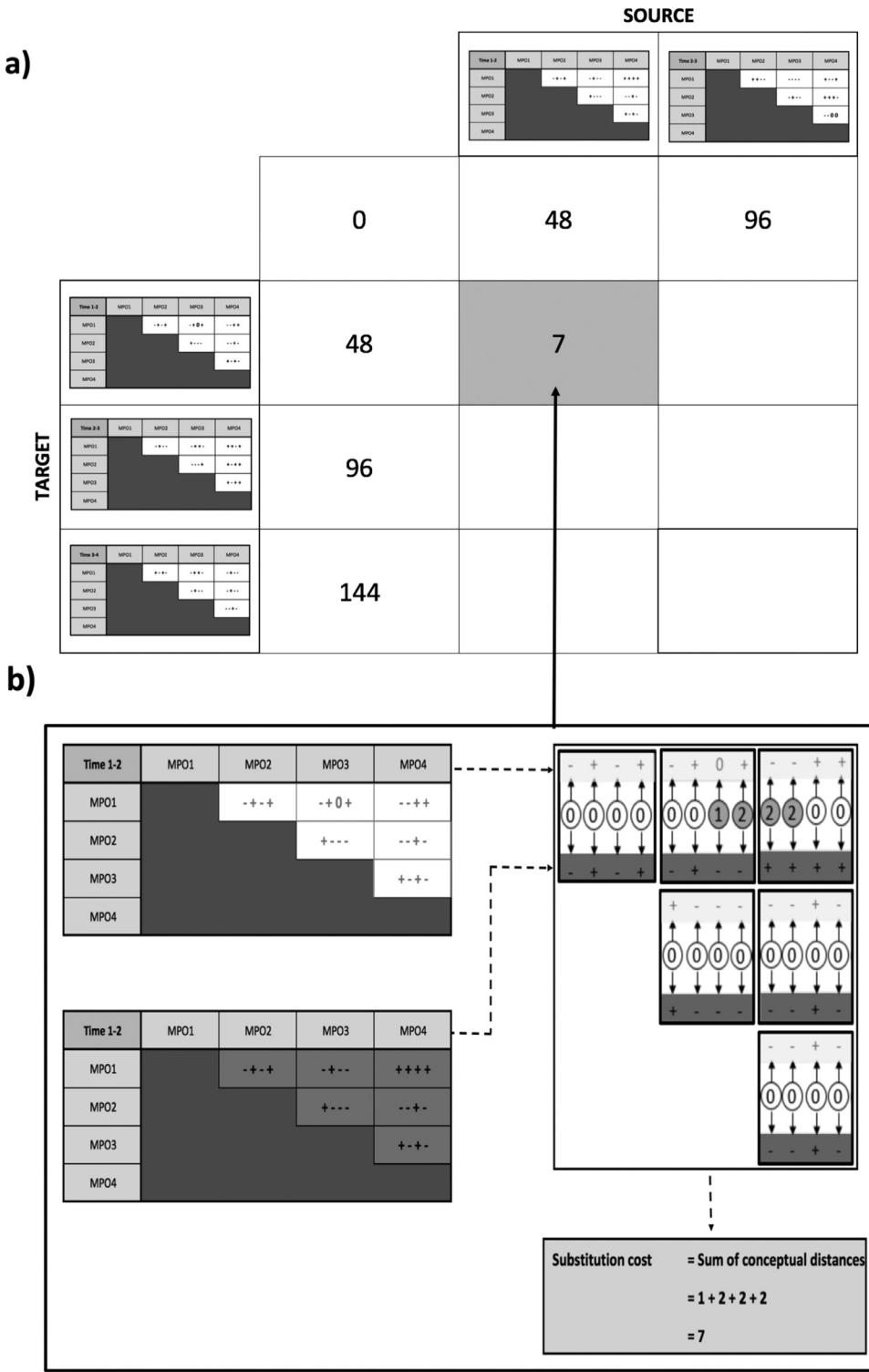


Fig. 9. Levenshtein distance calculation for QTC-matrix sequences of different length.

different shots. At this stage, we also want to stress that specific types of shots, such as a successful opening shot, can be achieved in multiple ways. It is, for example, also possible to first hit the yellow object ball instead of the red one (as was done in this example) to perform a successful opening shot.

4.4. Results

A distance matrix containing distances between each two shots is calculated by pairwise distance calculation. Each pair of QTC-matrix sequences is compared using the Levenshtein distance metric, allowing to detect similar movement patterns with (minor) temporal differences. The extent of the temporal differences that can be allowed for detecting movement patterns of different shots



Fig. 10. Starting formation of three cushion billiards.

is influenced by multiple factors, e.g. the overall similarity of the movements of the two shots, their lengths (the indel of one matrix will have a smaller impact on the resulting distance when comparing longer shots), the temporal resolution of the data or the distance threshold one defines for labelling two shots as similar. Based on this distance matrix, a hierarchical clustering (Fig. 12) was performed using the UPGMA (Unweighted Pair Group Method with Arithmetic Mean) method (Sokal & Michener, 1958). In the right part of the figure, it can be seen that, with some explainable outliers, all the shots are clustered according to their type. On the left of the hierarchical clustering, the trajectories of the different shots are visualised in an overlay, grouped by the type of shot.

To support the validation of these results, which is done by visual comparison of the ground truth (the type of the shot) with the hierarchical clustering, we show the trajectories of the two outliers of the clustering in Fig. 12. In Fig. 13, the shots '6 S' and '52 S', which are clustered as outliers with the 'R' and 'L' type, respectively, are shown. Although '6 S' is a successful shot (both the red and the yellow ball are hit, and the white ball hits three sides), it can be seen that the trajectories of the balls much more match the other shots of the 'R' type. The trajectories of the red and white balls in shot '52 S', on the other hand, much more match the trajectories of the balls of the 'L' type than those of the other successful shots. Compared to those shots, the yellow ball shows a rather limited displacement in the shot '52 S'. We can conclude that the hierarchical clustering indeed clusters the shots correctly, based on the trajectories of the balls in those shots. We are convinced that in this way, we supported the use of the proposed method for movement pattern detection of fragments of different length, containing movement of multiple MPOs.

To illustrate the alignment of the shots, we present the alignment and the distances of three shots ('28 R', '51 R' and '36 L') to the shot '23 R' in Fig. 14. It can be noted that shot '51 R' (represented by 101 QTC_C-matrices) is (slightly) more similar with shot '23 R' (98 QTC_C-matrices) than shot '28 R', which has a more similar number of QTC_C-matrices (100). Despite the higher number of deletions in the alignment of '51 R', it has a lower distance to '23 R' due to a higher number of identical QTC_C-matrices and less costly substitutions. The Levenshtein distance calculation method thus does not only allow to calculate distances between matrix sequences of different lengths, it also allows for the detection of similar spatiotemporal patterns that happen with different speeds.

Since both '28 R' and '51 R' have relatively low distances to '23R', a third alignment ('36 L') with a larger distance is included in Fig. 14. Because most of the shots in the dataset start in the same way but tend to differ at the end, most of the insertions and substitutions can be found at the end of the alignments.

5. Discussion

The method presented in this paper allows for calculating distances between matrix sequences of different length. In this paper, QTC was applied to a three cushion billiards case to exemplify the method. However, the proposed method can be applied on a broad range of phenomena that can be described by qualitative or quantitative calculi in the form of matrix sequences. Various applications can be seen in the domain of expert and intelligent systems as, for example, the analysis of movement interactions between humans and robots. While movement interactions between a single human and a single robot were already described by QTC (Hanheide et al., 2012), the proposed method enables the description and comparison of movement interactions between multiple humans and robots. In a world where robots are increasingly being used to replace humans for several tasks (Ajoudani et al., 2018), we are convinced that analysing and modelling of movement interactions between multiple humans and robots is an interesting research topic.

Another interesting application for expert and intelligent systems, is the automated analysis of traffic interactions at a microscopic level (i.e. analysing interactions between individual vehicles). Traffic analysis has seen a huge rise of interest in recent years, as traffic congestion is rising and self-driving cars are being developed and introduced in traffic. Below (Fig. 15) we present an example of two cars *m* and *l* overtaking a third car *k*, which was based on a simpler description of an overtake event by QTC (Van de Weghe et al., 2005). Two types of overtake can be discerned, i.e. a safe overtake, where car *m* waits with overtaking car *k* until car *l* has finished his overtaking of car *k*, and an unsafe overtake, where car *m* does not wait. The two types can happen at different speeds, illustrated by 5 variants in Fig. 15. The movements of the cars in those variants can be described by QTC_B (Van de Weghe, 2004) by matrix sequences of different length. Using the proposed method for comparing these QTC_B-matrix sequences, it is possible to group the different variants according to their type.

In geography, relations between MPOs or geographical regions (Egenhofer, 2010) can be described by means of topological relations (Klippe, Worboys, & Duckham, 2008). In this way, topological relations between MPOs or geographical regions can be stored in a matrix form for one specific timestamp. The movement of multiple objects in space can thus be described in a qualitative manner by a sequence of topological relation matrices, where the evolution of the pairwise topological relations for every combination of MPOs or geographical regions is described. The proposed distance calculation method can then be used for comparing the topological descriptions of movements of multiple MPOs or geographical regions during time intervals of different lengths. A conceptual distance table, containing all pairwise distances between the topological relations, is needed to calculate the indel and substitution costs that are used to compute the minimum edit distance.

Type	Group initial	Number	Average temporal length (seconds)	Stdev	Average temporal length (timestamps)	Stdev
success	S	14	5.01	0.07	100.14	1.35
pass on the right (+ yellow touched)	R	23	5.00	0.09	100.04	1.82
pass on the left	L	18	5.18	0.13	103.50	2.64

Fig. 11. Overview of the three cushion billiards dataset, with for each type of shot the average length in seconds and timestamps.

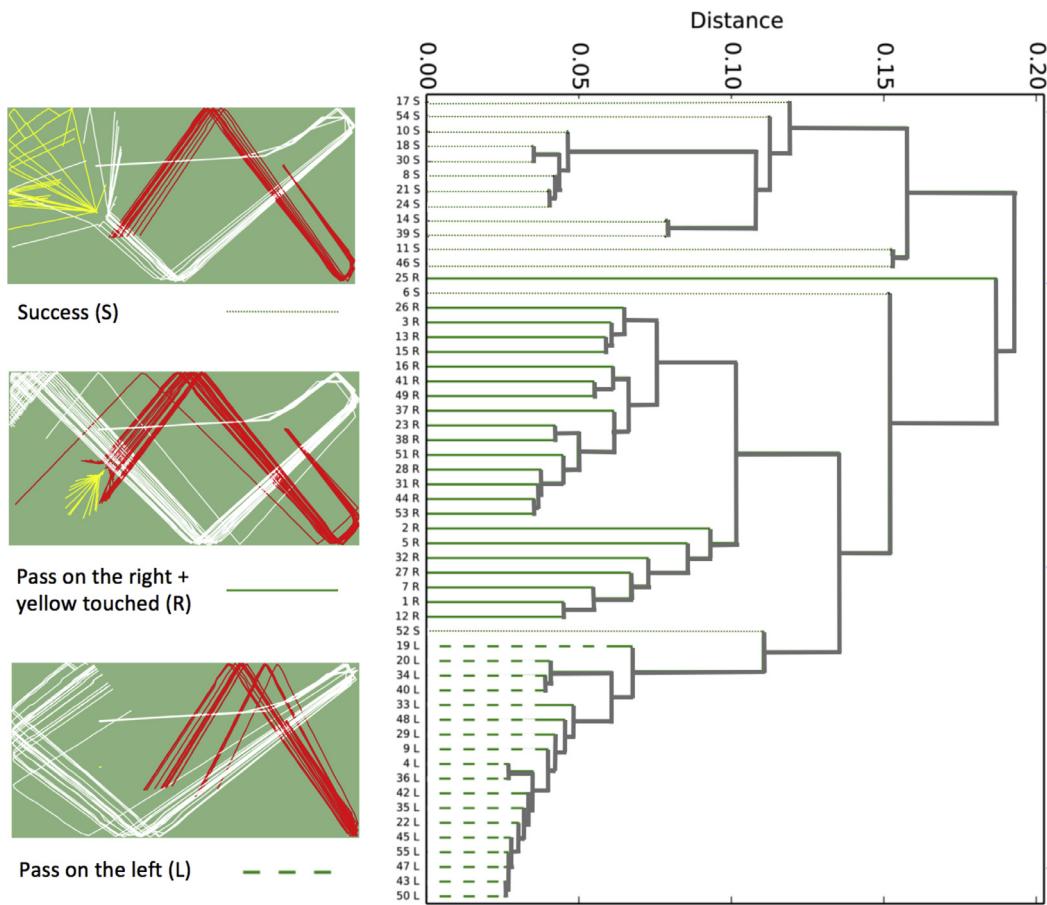


Fig. 12. Visualisations of the opening shots according to their type (left) and their hierarchical clustering (right) (Success = fine dotted line, Pass on the left = dotted line, Pass on the right & yellow touched = full line).

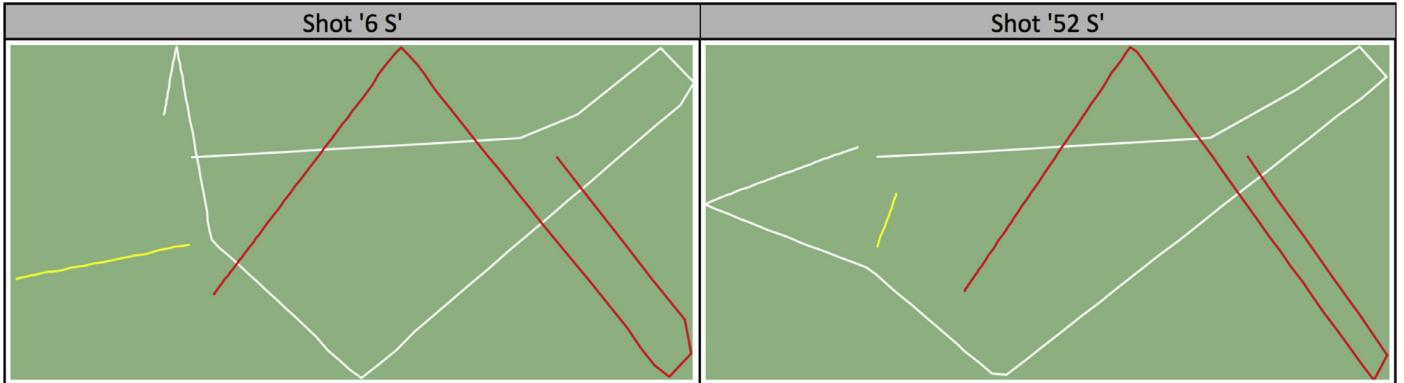


Fig. 13. Visualisations of the two shots '6 S' and '52 S' that were clustered as outliers.

Compared to established methods for distance calculation between sequences, the proposed method has some strengths and weaknesses. One of the advantages of using the proposed method with QTC, is the availability of conceptual distances (see Fig. 8), as opposed to other studies where the conceptual distance is either 0 or 1, based on thresholds for parameters such as the Euclidean distance (Chen et al., 2005; Ranacher & Tzavella, 2017). Secondly, the proposed method differs from these and other established methods in this domain, i.e. NWED (Dodge et al., 2012), DTW (Keogh & Pazzani, 2000) and EDR (Chen et al., 2005), by allowing for the comparison of fragments containing movement of multiple MPOs, which was stated to be an important research aim (see Section 3.2). Third, although not studied in-depth in this pa-

per, we do expect our method to be relatively accurate in calculating distances between matrix sequences that originate from noisy data. As mentioned by Dodge et al. (2012), and experimentally proven by Chen, Özsu, and Orià (2004) and Ding, Trajcevski, and Scheuermann (2008), distance calculations based on the edit distance (e.g. EDR, NWED) tend to be more accurate than distance calculations using the Euclidean distance or DTW in the presence of noise in the data of trajectories. Since our proposed method is based on the edit distance, we expect similar conclusions, although the impact of noise should be further investigated in the future.

An important limitation of the proposed methodology, however, is that it can only be used for comparing two matrix sequences with matrices having identical numbers of rows and columns. This

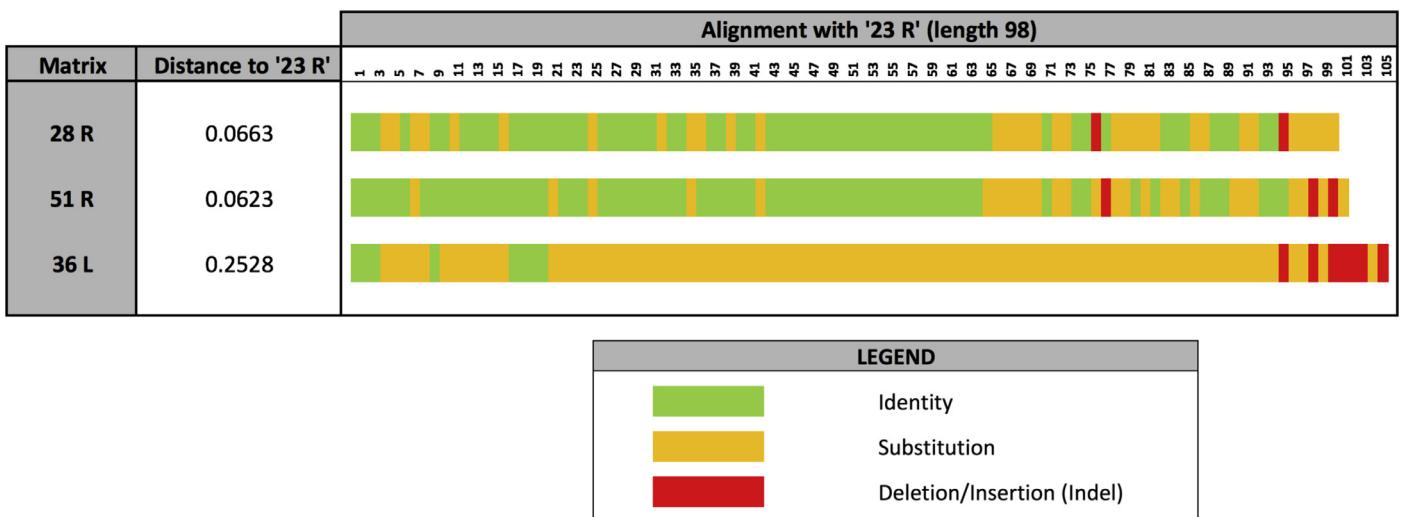


Fig. 14. Alignment of the three shots '28 R', '51 R' and '36 L' with the shot '23 R', and their resulting distances.

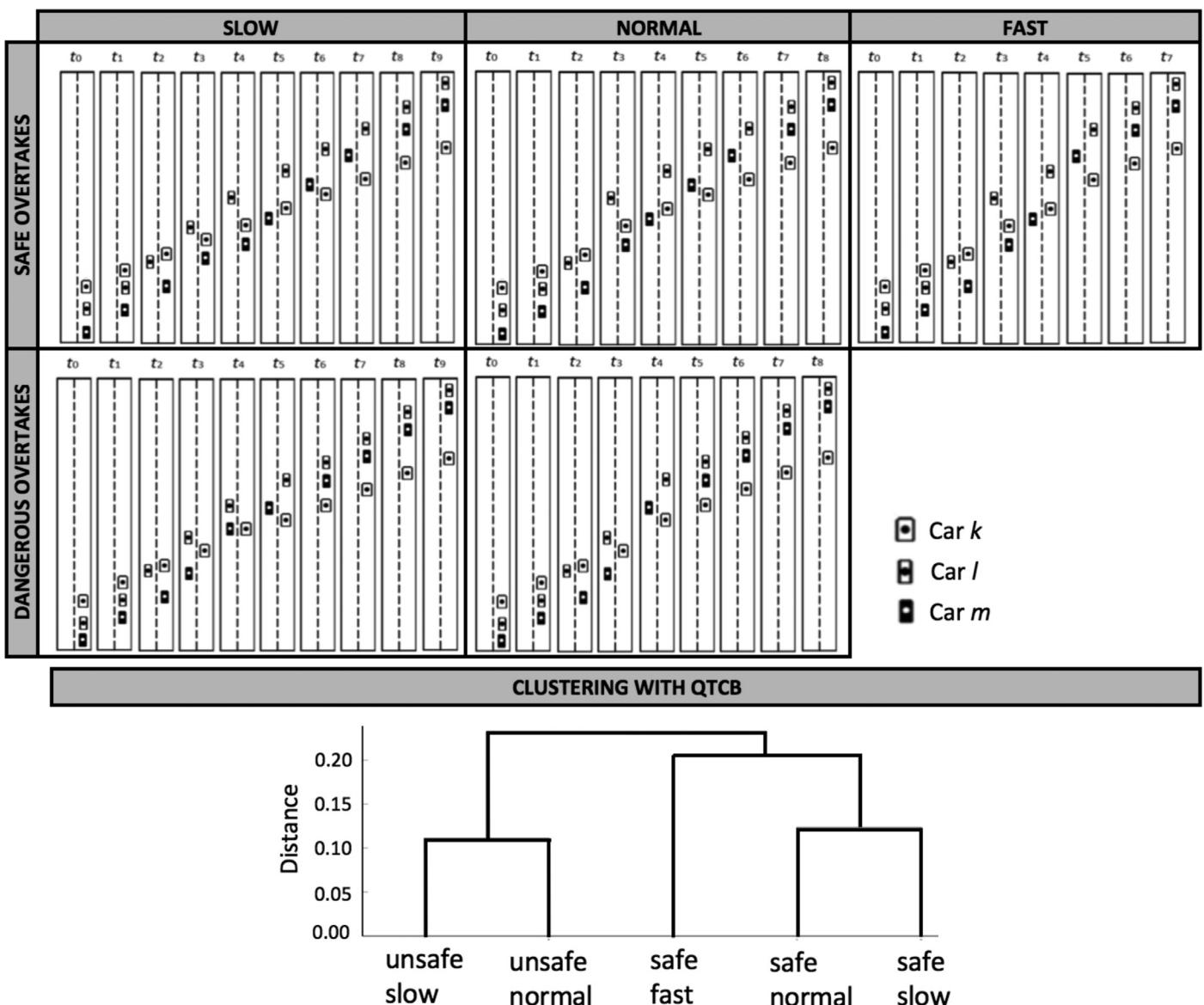


Fig. 15. Five overtake variants and their hierarchical clustering based on pairwise comparison of their QTC_{B2} matrix sequences.

means that the matrices contain relations between identical numbers of objects, thus having an identical number of cells. With the proposed method, it is therefore possible to compare phenomena of the same size, e.g. two spatial distributions of six cars on a car park or two variants of one car overtaking another. The comparison of phenomena containing different numbers of objects, which are usually more difficult to comprehend by human observers, are currently not supported. Another limitation is that calculating an edit distance matrix for every pair of compared matrix sequences requires high a computational power, especially when the matrix sequences are long or contain relations between a high number of MPOs.

6. Conclusion

This paper proposes a novel method for distance calculation between matrix sequences of different length, based on the Levenshtein distance metric. It was demonstrated that currently no optimal sequence alignment method for matrix sequences of different length is available. The proposed method enables, for example, the comparison of events of different duration that are expressed by means of matrix sequences. The matrix sequences can contain different elements from both qualitative and quantitative calculi, which in this paper was demonstrated by the application of the Qualitative Trajectory Calculus (QTC) to opening shots of three cushion billiards and a traffic analysis example. It must be stressed that the application is not limited to the described calculi in this paper. The method can be applied to a vast number of situations where matrix sequences need to be compared, whether they have the same number of matrices in each sequence or not.

Besides applying the method on different cases, future research should be aimed at analysing the impact of noise in the data on the distance calculation. Also, research could be done to extend the method to allow for distance calculation between matrix sequences containing matrices with different dimensions, thus enabling the comparison of movement fragments of not only different length, but also with different number of MPOs. Furthermore, the impact of the compression of the matrix sequences on the distance calculation, by for example eliminating redundant matrices, should be analysed and quantified. Other future research could investigate the effects of speed differences of identical trajectories on the distance calculation, and confront the results with established methods such as DTW.

Acknowledgements

The authors would like to thank the Research Foundation – Flanders (FWO) for funding the research of Jasper Beernaerts. We also want to show our gratitude to Gang Wang from the Department of Mathematical Modelling, Statistics and Bio-informatics of Ghent University, for generating the three cushion billiards dataset.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at [doi:10.1016/j.eswa.2018.07.076](https://doi.org/10.1016/j.eswa.2018.07.076).

References

- Abbott, A. (1995). Sequence analysis: New methods for old ideas. *Annual Review Sociology*, 21(1), 93–113.
- Abbott, A., & Tsay, A. (2000). Sequence analysis and optimal matching methods in Sociology. *Sociological Methods & Research*, 29(1), 3–33.
- Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., & Khatib, O. (2018). Progress and prospects of the human-robot collaboration. *Autonomous Robots*, 42(5), 957–975.
- Allen, J. F. (1981). An interval-based representation of temporal knowledge. In J. F. Allen (Ed.), *Proceedings 7th IJCAI* (pp. 221–226). Morgan Kaufmann Publishers.
- Bera, R., & Claramunt, C. (2003). Topology-base proximities in spatial systems. *Journal of Geographical Systems*, 5(4), 353–379.
- Billen, R., & Zlatanova, S. (2003). 3D Spatial relationships model: A useful concept for 3D cadastre? *Computers, Environment and Urban Systems*, 27, 411–425.
- Byrne, R. (1998). *Byrne's new standard book of pool and billiards*. San Diego: Harcourt.
- Chavoshi, S. H., De Baets, B., Neutens, T., De Tre, G., & Van de Weghe, N. (2015). Exploring dance movement data using sequence alignment methods. *PLoS ONE*, 10(7), 1–25.
- Chen, L., Özsu, M. T., & Oria, V. (2004). Symbolic representation and retrieval of moving object trajectories. In *Proceedings of the 6th ACM SIGMM international workshop on multimedia information retrieval* (pp. 227–234). ACM.
- Chen, L., Özsu, M. T., & Oria, V. (2005). Robust and fast similarity search for moving object trajectories. In *Proceedings of the ACM SIGMOD international conference on management of data* (pp. 491–502). ACM.
- Cohen, G. L. (2002). Three cushion billiards: Notes on the diamond system. *Sports Engineering*, 5(1), 43–51.
- Cohn, A. G., Bennett, B., Gooday, J., & Gotts, N. (1997). Qualitative spatial representation and reasoning with the region connection calculus. *Geoinformatica*, 1(3), 1–44.
- Cohn, A. G. (2005). Qualitative spatial representation and reasoning techniques. *Lecture Notes in Computer Science*, 1303(1), 1–30.
- Delafontaine, M., Cohn, A. G., & Van de Weghe, N. (2011). Implementing a qualitative calculus to analyse moving point objects. *Expert Systems with Applications*, 38(5), 5187–5196.
- Delafontaine, M., Chavoshi, S. H., Cohn, A., & Van de Weghe, N. (2012). A Qualitative Trajectory Calculus to reason about moving objects. In M. S. Hazarika (Ed.), *Qualitative spatio-temporal representation and reasoning: Trends and future directions* (pp. 148–168). Hershey: IGI Global.
- Delafontaine, M., Versichele, M., Neutens, T., & Van de Weghe, N. (2012). Analysing spatiotemporal sequences in Bluetooth tracking data. *Applied Geography*, 34, 659–668.
- Dijkmeijer, J., & de Hoop, S. (1996). Topologic relations between fuzzy area objects. In M. J. Kraak, & M. Molenaar (Eds.), *Advances in GIS research II, proceedings of the 7th spatial data handling* (pp. 377–394). Taylor & Francis.
- Ding, H., Trajcevski, G., & Scheuermann, P. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proceedings of VLDB Endowment*, 1(2), 1542–1552.
- Dodge, S., Laube, P., & Weibel, R. (2012). Movement similarity assessment using symbolic representation of trajectories. *International Journal of Geographical Information Science*, 26(9), 1563–1588.
- Egenhofer, M. J. (1989). A formal definition of binary topological relationships. *Lecture Notes in Computer Science*, 367, 457–472.
- Egenhofer, M. J., & Herring, J. R. (1991). *Categorizing binary topologic relations between regions, lines and points in geographic databases*. Orono: Dept. of Surveying Engineering, University of Maine Tech. Report.
- Egenhofer, M. J., & Shariff, A. R. (1998). Metric details for natural-language spatial relations. *ACM Transactions on Information Systems*, 16(4), 295–321.
- Egenhofer, M. J. (2010). The family of conceptual neighborhood graphs for region-region relations. *Geographic Information Science*, 6292, 42–55.
- Galton, A. (2001). Dominance diagrams: A tool for qualitative reasoning about continuous systems. *Fundamenta Informaticae*, 46(1), 55–77.
- Galton, A. (1990). A critical examination of Allen's theory of action and time. *Artificial Intelligence*, 42(2), 159–188.
- Gentle, J. E. (2007). *Matrix algebra: Theory, computations, and applications in statistics*. Berlin: Springer.
- Hanheide, M., Peters, A., & Bellotto, N. (2012). Analysis of human-robot spatial behaviour applying a qualitative trajectory calculus. In *The 21st IEEE international symposium on robot and human interactive communication* (pp. 689–694). Curran Associates.
- Ibrahim, Z. M., & Tawfik, A. Y. (2007). An abstract theory and ontology of motion based on the regions connection calculus. In I. Miguel, & W. Ruml (Eds.), *Abstraction, reformulation, and approximation* (pp. 230–242). Whistler: Springer.
- Keogh, E., & Pazzani, M. (2000). Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 285–289). ACM.
- Klippel, A., Worboys, M., & Duckham, M. (2008). Identifying factors of geographic event conceptualization. *International Journal of Geographical Information Science*, 22(2), 183–204.
- Kurata, Y., & Egenhofer, M. J. (2009). Interpretation of behaviours from a viewpoint of topology. In B. Gottfriend, & H. Aghajan (Eds.), *Behaviour monitoring and interpretation-ambient intelligence and smart environments* (pp. 75–97). Amsterdam: IOS Press.
- Kwan, M. P., Xiao, N., & Ding, G. (2014). Assessing activity pattern similarity with multidimensional sequence alignment based on multiobjective optimization evolutionary algorithm. *Geographical Analysis*, 46(3), 297–320.
- Laube, P., Van Kreveld, M., & Imfeld, S. (2004). Finding REMO—detecting relative motion patterns in geospatial lifelines. In P. F. Fisher (Ed.), *Developments in spatial data handling, proceedings of the 11th international symposium on spatial data handling* (pp. 201–214). Springer.
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8), 707–710.
- Morisson, D. A. (2010). Sequence alignment: Methods, models, concepts, and strategies. *Systematic Biology*, 59, 363–365.
- Muller, P. (2002). Topological spatio-temporal reasoning and representation. *Computational Intelligence*, 18(3), 420–450.

- Needleman, S. B., & Wunch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3), 443–453.
- Ranacher, P., & Tzavella, K. (2017). How to compare movement? A review of physical movement similarity measures in geographic information science and beyond. *Cartography and Geographic Information Science*, 41(3), 286–307.
- Randell, D., Cui, Z., & Cohn, A. G. (1992). A spatial logic based on regions and connection. In B. Nebel, W. Swartout, & C. Rich (Eds.), *Proceedings of the 3rd international conference on knowledge representation and reasoning* (pp. 165–176). Morgan Kaufman.
- Rosenberg, M. S. (2009). *Sequence alignment: methods, models, concepts, and strategies*. Berkeley, CA: University of California Press.
- Schlüch, R. (2003). Homogenous groups of travellers. In *10th International conference on travel behaviour research* (pp. 10–15).
- Shoval, N., & Isaacson, M. (2007). Sequence alignment as a method for human activity analysis in space and time. *Annals of the Association of American Geographers*, 97(2), 282–297.
- Sokal, R., & Michener, C. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38, 1409–1438.
- Stehle, S., & Peuquet, D. J. (2015). Analyzing spatio-temporal patterns and their evolution via sequence alignment. *Spatial Cognition & Computation: An Interdisciplinary Journal*, 15(2), 68–85.
- Tönges, U., Perrey, S. W., Stoye, J., & Andreas, W. M. A. (1996). A general method for fast multiple sequence alignment. *Gene*, 172(1), 33–41.
- Van de Weghe, N. (2004). *Representing and reasoning about moving objects: A qualitative approach*. Ghent University.
- Van de Weghe, N., & De Maeyer, P. (2005). Conceptual neighbourhood diagrams for representing moving objects. *Lecture Notes in Computer Science*, 3770, 228–238.
- Van de Weghe, N., Cohn, A. G., De Maeyer, P., & Witlox, F. (2005). Representing moving objects in computer-based expert systems: The overtaking event example. *Expert Systems with Applications*, 29(4), 977–983.
- Van de Weghe, N., Cohn, A. G., De Tré, G., & De Maeyer, P. (2006). A Qualitative Trajectory Calculus as a basis for representing moving objects in Geographical Information Systems. *Control and Cybernetics*, 35(1), 97–120.
- Waterman, M. S. (1984). General methods of sequence comparison. *Bulletin of Mathematical Biology*, 46(4), 473–500.
- Wilson, C. (2006). Reliability of sequence-alignment analysis of social processes: Monte Carlo tests of ClustalG software. *Environment and Planning A*, 38, 187–204.
- Wilson, C. (2008). Activity patterns in space and time: Calculating representative Hagerstrand trajectories. *Transportation*, 35, 485–499.
- Worboys, M. F. (1994). A Unified model for spatial and temporal information. *The Computer Journal*, 37(1), 26–34.