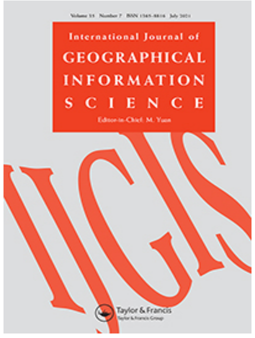


International Journal of Geographical Information Science

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tgis20

The Point-Descriptor-Precedence representation for point configurations and movements

Amna Qayyum, Bernard De Baets, Muhammad Sulman Baig, Frank Witlox, Guy De Tré & Nico Van de Weghe

To cite this article: Amna Qayyum, Bernard De Baets, Muhammad Sulman Baig, Frank Witlox, Guy De Tré & Nico Van de Weghe (2021) The Point-Descriptor-Precedence representation for point configurations and movements, International Journal of Geographical Information Science, 35:7, 1374-1391, DOI: 10.1080/13658816.2020.1864378

To link to this article: https://doi.org/10.1080/13658816.2020.1864378

Published online: 11 Jan 2021.

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
2021, VOL. 35, NO. 7, 1374–1391
https://doi.org/10.1080/13658816.2020.1864378

RESEARCH ARTICLE

The Point-Descriptor-Precedence representation for point configurations and movements

Amna Qayyum [a], Bernard De Baets [b], Muhammad Sulman Baig [a], Frank Witlox [c], Guy De Tré [d] and Nico Van de Weghe [a]

[a]CartoGIS, Department of Geography, Ghent University, Ghent, Belgium; [b]KERMIT, Department of Data Analysis and Mathematical Modelling, Ghent University, Ghent, Belgium; [c]SEG, Department of Geography, Ghent University, Ghent, Belgium; [d]DDCM, Department of Telecommunications and Information Processing, Ghent University, Ghent, Belgium

ABSTRACT

In this paper, we represent (moving) point configurations along a curved directed line qualitatively by means of a system of relational symbols based on two distance descriptors: one representing distance along the curved directed line and the other representing signed orthogonal distance to the curved directed line. The curved directed line represents the direction of the movement of interest. For instance, it could be straight as in the case of driving along a highway or could be curved as in the case of an intersection or a roundabout. Inspired by the Point Calculus, the order between the points on the curved directed line is described by means of a small set of binary relations ($<, =, >$) acting upon the distance descriptors. We call this representation the *Point-Descriptor-Precedence-Static* (PDP_S) representation at a time point and *Point-Descriptor-Precedence-Dynamic* (PDP_D) representation during a time interval. To illustrate how the proposed approach can be used to represent and analyse curved movements, some basic micro-analysis traffic examples are studied. Finally, we discuss some extensions of our work to highlight the practical benefits of PDP in identifying motion patterns that could be useful in GIS, autonomous vehicles, sports analytics, and gait analysis.

ARTICLE HISTORY
Received 5 May 2020
Accepted 11 December 2020

KEYWORDS
Qualitative representation; curved directed line; spatiotemporal data representation; traffic research; pattern detection

1. Introduction

Qualitative spatiotemporal representations provide an intuitive way of modelling the most relevant facets of space and time for a particular task, whether it relates to designing intelligent systems or logical deductions for cognitive computing. The way how 'objects' are perceived in such representations depends on the context of the application. For instance, certain applications not involving the mass and other dimensions of objects consider objects as 'points' to simplify calculations; others consider them as lines or regions. Since object motion is centric in many of such representations, significant research has been carried out to conceptualize movements following a straight path. However, curved path movements are equally important since not all movements tend to follow a straight path in the real life just like movement at road intersections, roller

CONTACT Amna Qayyum amna.qayyum@ugent.be

coasters, motion of a basketball into the basket, and the earth's rotation around the sun. To the best of our knowledge, we are not aware of any such representation that can describe curved movements in a specific direction and can be used for pattern detection, which is the primary goal of our work. We use 'points' to represent objects moving along a curved line in a specific direction and use distance descriptors to describe the order between their positions qualitatively with respect to the curved line.

The concept of ordering between points was first introduced by Vilain and Kautz (1986) in their *Point Calculus*, which represents time points along a line defined by the constraints ($<, =, >$) and their combinations ($\leq, \neq, \geq$) in 1-D. The *n*-D *Point Calculus* was introduced by Balbiani and Condotta (2002) from an AI-theoretical perspective to reason about points across the *n*-D Euclidean space. The idea of spatial reasoning by using cardinal directions was introduced by Frank (1996), whereas relative directions were explored in the *Double-Cross Calculus* by Zimmermann and Freksa (1996). Then came the binary-*Cyclic Calculus* (CYC_b) (Isli and Cohn 2000), which symbolically represents four base relations (equal, left, right and opposite) between the directions of the points in a 2-D plane, along with their union and intersection. The ternary-CYC (CYC_t), which operates on a set of (ternary) relations on 2-D orientations, was an important breakthrough in the field of ternary calculi and is more expressive than CYCb. However, ternary calculi are complex and difficult to interpret from a cognitive perspective (Condotta *et al*. 2006).

In what followed, Skiadopoulos and Koubarakis (2001) used cardinal directions for locating spatial regions along the *x*- and *y*-axes. Their work was extended into the *Cardinal Direction Relations calculus* (CDR), which places the regions in corresponding tiles of a bounded 2-D frame of reference (Skiadopoulos and Koubarakis 2005). Another direction calculus that deals with the relative locations of the domain entities is the *Oriented Point Relation Algebra* (OPRA) devised by Moratz (2006) and extended to multiple directions by Dylla and Lee (2010) and Mossakowski and Moratz (2012). Note that these approaches do not allow to represent curved movements in a specific direction.

Kurata and Shi (2008) modelled regions based on their relative directions as *Region-in-the-frame-of-Directed-Line* (RfDL-3-12). Van de Weghe *et al*. (2005) and Glez-Cabrera *et al*. (2013) targeted traffic modelling at the microscale and established the possible relations of a point (vehicle) with respect to the trajectory of another point depending on the cross-point of the trajectories. This work takes into account the direction of moving points in a 2-D plane but is ill-defined if one of the vehicles is at rest. Li and Liu (2015) represented the cardinal directions of two regions by direction-relation matrices. Arrufi and Kirsch (2018) described the motion of points (vehicles) in a Cartesian frame only by using their velocity representations, thereby ignoring relative directions. In addition, there is no direct link between the *Qualitative Trajectory Calculus on Networks* (QTC_N) given by Delafontaine *et al*. (2011) and our approach, since QTC_N uses a single dimension only (*i. e*., the shortest path distance) to represent objects moving along a network, making it impossible to conduct micro-traffic analyses. QTC_N is thus interesting for macro-traffic analyses, whereas our approach is interesting for micro-traffic analyses as we describe objects with respect to a specific curved movement direction by means of two distance descriptors and it is possible to combine several descriptors.

Every spatiotemporal approach listed above has its strong points as well as its limitations. Not all of them take traffic vehicles as their moving objects, yet traffic has remained central in the majority of representations. Besides, the traditional approaches either focus on the cardinal directions or the relative directions to locate moving entities. Whereas in PDP, the direction of the movement of interest is dependent upon the context of the application. Apart from representing points along a straight line, PDP is capable to handle curved movements in any spatial arrangement. In order to illustrate this, we will present some basic traffic micro-analysis examples where the moving objects (vehicles) are described in terms of their precedence given by two distance descriptors: one in the driving direction, which is represented as a parametrized curved directed line and one in an orthogonal direction to the curved directed line. In an extension, we added a third descriptor to differentiate between a safe and a dangerous overtake movement pattern. It is pertinent to mention here that the proposed approach determines the precedence between the points by using the path subtype of the extrinsic frame of reference (Clementini 2013).

This paper is organized as follows: In Section 2, the basics of our proposed approach are introduced. Section 3 illustrates PDP using real-world traffic situations. Section 4 extends the basics of PDP with various traffic examples showing that it holds a rich potential for many extensions. Finally, our findings are discussed and summarized along with some possible future extensions.

2. Defining the Point-Descriptor-Precedence (PDP) representation

2.1. Conventions

In the *Point-Descriptor-Precedence* (PDP) representation, we start with n (moving) points (objects) $o_1, o_2, \ldots, o_n$ in an m-D space. Throughout this paper, we use 'Points' as our spatial entities. We then select the movement of interest, which is parametrized and can be a straight or curved directed line depending upon the context of the application. With respect to this directed line, we construct s descriptors $d_1, d_2, \ldots, d_s$, which mainly describe the distance of points travelled along and away from the directed line. Next, we order the n points in terms of their qualitative distances described by the descriptors by using a system of relational symbols ($<, =, >$) and refer to it as descriptor-precedence (*i.e.*, d_1-precedence, d_2-precedence, … , d_s-precedence).

Figure 1 illustrates the idea of PDP by showing two examples of a directed line: straight and curved. Here, the first descriptor d_1 describes the distance of points travelled along the directed line and the second descriptor d_2 describes the signed orthogonal distance of the points to the directed line measured positive on the left, and negative on the right with respect to the direction. It is pertinent to mention here that if the directed line represents a movement along the x-axis, the two distance descriptors are just the coordinates in the Euclidean system for which the ordering relations between the points have already been explored in the n-D *Point Calculus* given by Balbiani and Condotta (2002). Hence, the n-D *Point Calculus* is just a specialized case of PDP suitable for representing points along a straight line. However, the spatial reasoning concepts described in the *Point Calculus* are in no way related to PDP.

Sections 2.2 and 2.3 outline the two types of PDP representations: static and dynamic, with the help of some basic micro-traffic examples.

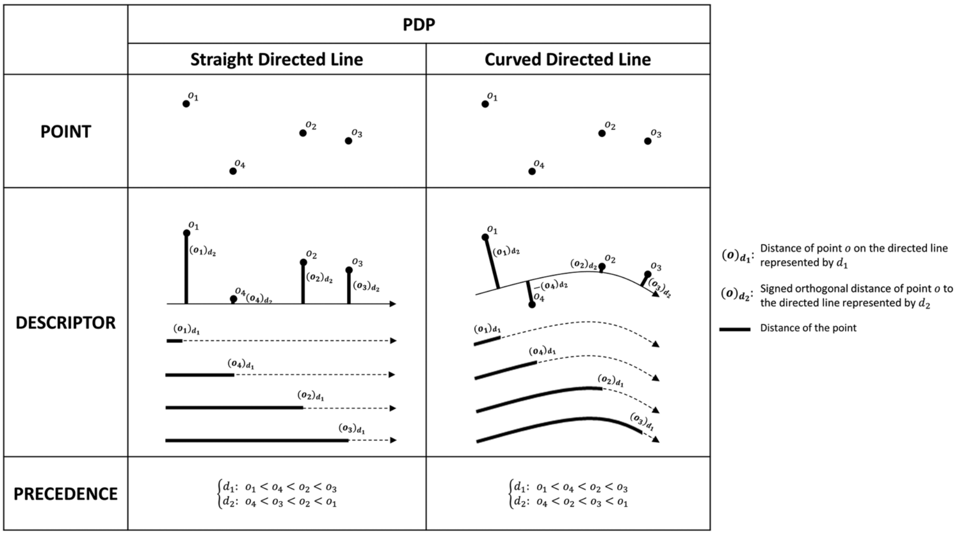


Figure 1. PDP (Point-Descriptor-Precedence) representation for straight and curved directed lines.

2.2. Point-Descriptor-Precedence-Static (PDP_S) representation

PDP_S describes point configurations at specific time points. For illustration purposes, consider a simple overtake movement pattern depicted in Figure 2(a) at time points $t_1 < t_2 < t_3 < t_4 < t_5$. Here, vehicle *l* is overtaking vehicle *k* on a two-lane road. The two distance descriptors are shown in Figure 2(b), where the first descriptor d_1 represents the distance travelled by the vehicles along the driving direction taken at the centre of the road and the second descriptor d_2 represents the signed orthogonal distance of the vehicles from d_1.

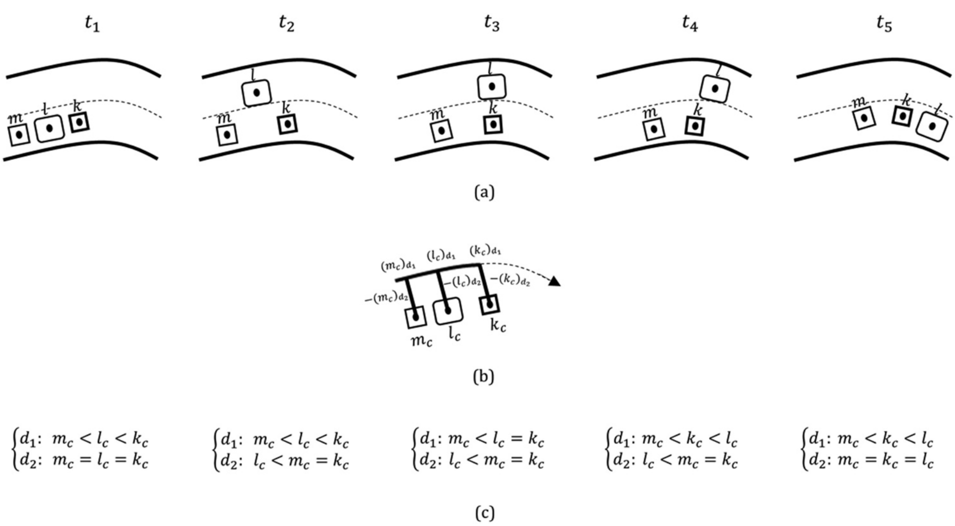


Figure 2. An overtake movement pattern comprised of three vehicles k, l and m on a two-lane road (a), two distance descriptors: d_1 taken along the centre of the road and d_2 is the signed orthogonal distance of the vehicles from d_1 (b), PDP_S representation using the centroid of each vehicle (c).

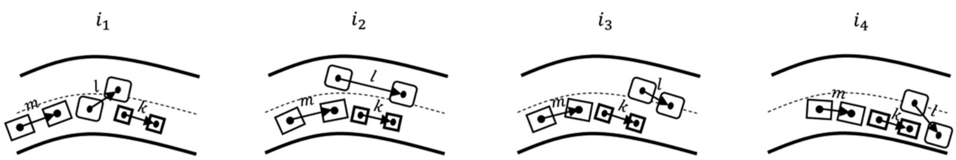
Figure 2(c) shows the corresponding precedence of the centroids 'c' of the vehicles at every time point by means of the two descriptors.

The usefulness of PDP_S will be elaborated later in Section 3 of this paper.

2.3. Point-Descriptor-Precedence-Dynamic (PDP_D) representation

Unlike PDP_S where the points (vehicles) are represented at time points, PDP_D symbolizes points during time intervals. For that purpose, the velocity vector representing the rate of change of the position of each point (vehicle) is captured and it is possible to see the vehicles' driving behaviour during these intervals. For instance, consider the overtake movement pattern shown in Figure 3(a). Figure 3(b) represents the precedence of vehicles k, l and m during every time interval corresponding to the initial state (denoted by '$-$' sign) and final state (denoted by '$+$' sign) of each vehicle (hence defining the velocity vector).

By using the velocity vector, PDP_D can also be used to represent bi-directional traffic motion in adjacent lanes. For instance, consider four motion scenarios of bi-directional traffic on adjacent lanes during different intervals (i_e, i_f, i_g and i_h) in Figure 4. By analysing the velocity vector of both vehicles (k and l) given by the distance descriptors (d_1 and d_2),

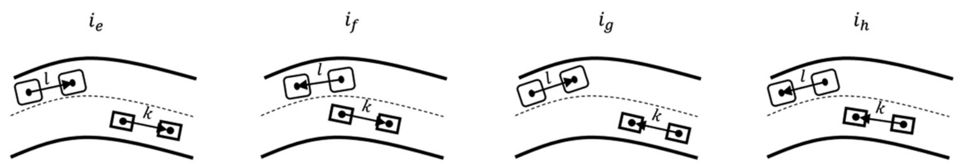


(a)

$\begin{cases} d_1: m_c^- < m_c^+ < l_c^- < l_c^+ < k_c^- < k_c^+ \\ d_2: l_c^+ < m_c^- = m_c^+ = l_c^- = k_c^- = k_c^+ \end{cases}$ $\begin{cases} d_1: m_c^- < m_c^+ = l_c^- < k_c^- < k_c^+ < l_c^+ \\ d_2: l_c^- = l_c^+ < m_c^- = m_c^+ = k_c^- = k_c^+ \end{cases}$ $\begin{cases} d_1: m_c^- < m_c^+ < k_c^- < l_c^- = k_c^+ < l_c^+ \\ d_2: l_c^- < l_c^+ < m_c^- = m_c^+ = k_c^- = k_c^+ \end{cases}$ $\begin{cases} d_1: m_c^- < m_c^+ < k_c^- < l_c^- = k_c^+ < l_c^+ \\ d_2: l_c^- < m_c^- = m_c^+ = k_c^- = k_c^+ = l_c^+ \end{cases}$

(b)

Figure 3. An overtake movement pattern comprised of three vehicles k, l and m (a) shown using the PDP_D representation of the centroid of each vehicle (b).



(a)

$\begin{cases} d_1: l_c^- < l_c^+ < k_c^- < k_c^+ \\ d_2: l_c^- = l_c^+ < k_c^- = k_c^+ \end{cases}$ $\begin{cases} d_1: l_c^+ < l_c^- = k_c^- < k_c^+ \\ d_2: l_c^+ = l_c^- < k_c^- = k_c^+ \end{cases}$ $\begin{cases} d_1: l_c^- < l_c^+ < k_c^+ < k_c^- \\ d_2: l_c^- = l_c^+ < k_c^+ = k_c^- \end{cases}$ $\begin{cases} d_1: l_c^+ < l_c^- < k_c^+ < k_c^- \\ d_2: l_c^+ = l_c^- < k_c^+ = k_c^- \end{cases}$

(b)

Figure 4. Four motion scenarios of bi-directional traffic during different intervals (a) with the corresponding PDP_D representations (b).

the change in the vehicles' heading direction can be easily observed in the corresponding precedence (Figure 4(b)).

3. Applying PDP to traffic analysis

PDP can be used to identify movement patterns in a given motion dataset. How this can be implemented mainly depends on whether we use the interval-based approach, the event-based approach, or a combination of both approaches to model spatiotemporal data. In the event-based approach, the time associated with each change (*i.e.*, each event) is stored in an increasing order from the initial 'state' at a time point t_0 to the latest recorded change at a time point t_n (Peuquet and Duan 1995). Similarly, the interval-based approach or, strictly speaking, the time-based approach represents an ordered progression of known changes on a timeline (Peuquet and Duan 1995). Typically in this interval-based approach, the time intervals have the same length (e.g., the temporal resolution of a tracking device). For applying PDP in real-world traffic analysis, we have used a combination of both event-based and equal-interval approaches to extract a specific movement pattern from a particular traffic scenario as illustrated in this section.

3.1. Dataset

We started with the analysis of a series of images extracted from a video (Kumar 2018) that contains a top-view of traffic on a busy three-lane highway in downtown Los Angeles, USA. The duration of the video is 9.6 seconds. We applied two distance descriptors d_1 and d_2 on the video: d_1 describes the distances travelled by the vehicles in the driving direction and d_2 describes the signed orthogonal distances of the vehicles to the driving direction. The video has been rotated 180° clockwise, and for reference, only the first and the last frames of the video are displayed in grey-scale, respectively, in Figures 5 and 6. For our current analysis, we only considered 16 vehicles labelled *a*–*p*. The centroids of these 16 vehicles were logged in *x*-, *y*- and *t*-coordinates at a temporal resolution of 10 Hz. By adjusting the temporal resolution of the dataset, we implemented an equal-interval-based approach in our procedure.

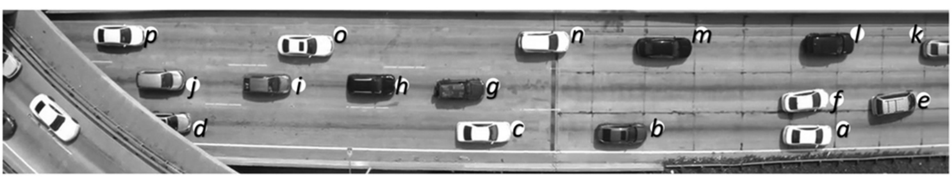


Figure 5. The first frame of the video.

Figure 6. The last frame of the video.

The selected video is an example of a movement along a straight directed line that has been chosen as parallel to the driving direction. Though we did not see a lane-changing event, there were patterns where a vehicle passed some other vehicles on its right. This represented a change in the precedence of the vehicles described by d_2 whereas the precedence of vehicles described by d_1 remained the same. Therefore, our main objective was to identify which vehicles in our selected sample of 16 vehicles displayed this particular movement pattern (passing some other vehicles on the right). For the ease of referencing throughout our process, we call this a pass movement pattern.

3.2. Pre-processing of the dataset

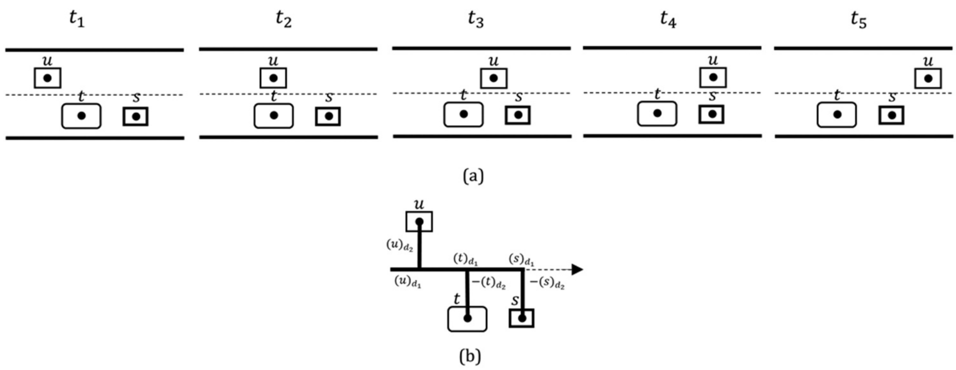
As discussed in Section 3.1, apparently the vehicles represented a change in their d_1-precedence only for exhibiting a pass movement pattern. Nonetheless, we arranged 16 vehicles in their corresponding d_1-precedence and d_2-precedence for each frame and recorded the changes in the precedence of the vehicles given by each distance descriptor. The first five changes in the d_1- and d_2-precedence of the vehicles are presented in Table 1. Note that the changes (highlighted in bold) occurred at a specific time point (frame) in the dataset and we recorded only these time points to make a new dataset out of the original one. We refer to this as pre-processing of our original dataset, which is consistent with the event-based approach where the changes occurring at certain time points are recorded. This pre-processing is important for removing the redundant information from the main dataset (*i.e.*, where the precedence remains unchanged). Technically, this also reduced the calculation time for our analysis and the original dataset was compressed from 102 time points to 32 time points only. Moreover, the equality relations between the vehicles were derived within a threshold of ± 0.5m meaning that if the difference between the vehicles' coordinates was found within the range of $(-0.5, +0.5)$, the positions of the vehicles were considered as equal while deriving precedences using the respective descriptor.

3.3. Construction of a reference movement pattern

After pre-processing the dataset, we constructed a reference movement pattern of three vehicles s, t and u that describes the pass movement pattern introduced in Section 3.1.

Table 1. First five changes in the precedence of vehicles recorded at specific time points (frames).

Time point	Precedence
t_1	$\begin{cases} d_1: p<j<d<i<o<h<g<c<n<b<a=f<l<e<k \\ d_2: k=l=m=n=o=p<e=f=g=h=i=j<a=b=c=d \end{cases}$
t_2	$\begin{cases} d_1: p<j<d<i<o<h<g<c<n<b<f<a<l<e<k \\ d_2: k=l=m=n=o=p<e=f=g=h=i=j<a=b=c=d \end{cases}$
t_8	$\begin{cases} d_1: p<j<d<i<o<h<g<c<n<b<f<a=l<e<k \\ d_2: k=l=m=n=o=p<e=f=g=h=i=j<a=b=c=d \end{cases}$
t_{10}	$\begin{cases} d_1: p<j<d<i<o<h<g<c<n<b<f<l<a<e<k \\ d_2: k=l=m=n=o=p<e=f=g=h=i=j<a=b=c=d \end{cases}$
t_{16}	$\begin{cases} d_1: p<j<d<i<o<h<g<c<n<b<f<l<a=e<k \\ d_2: k=l=m=n=o=p<e=f=g=h=i=j<a=b=c=d \end{cases}$
t_{17}	$\begin{cases} d_1: p<j<d<i<o<h<g<c<n<b<f<l<e<a<k \\ d_2: k=l=m=n=o=p<e=f=g=h=i=j<a=b=c=d \end{cases}$



$$\begin{cases} d_1: & u < t < s \\ d_2: & u < s = t \end{cases} \quad \begin{cases} d_1: & u = t < s \\ d_2: & u < s = t \end{cases} \quad \begin{cases} d_1: & t < u < s \\ d_2: & u < s = t \end{cases} \quad \begin{cases} d_1: & t < u = s \\ d_2: & u < s = t \end{cases} \quad \begin{cases} d_1: & t < s < u \\ d_2: & u < s = t \end{cases}$$

(c)

Figure 7. Reference movement pattern of three vehicles u, s and t using PDP_S representation. u is passing t and s on its right.

Vehicle u is passing two vehicles t and s. Our reference movement pattern was logged in x- and y-coordinates at five time points (Figure 7). We represented this reference movement pattern using the PDP_S representation. Each of the five time points represents a change in the d_1-precedence of the vehicles. This was done deliberately since we have adopted an event-based approach where the change in the sequence of events is important. At t_1: u is behind t and s; at t_2: u comes adjacent to t but is still behind s; at t_3: u crosses t; at t_4: u comes adjacent to s and at t_5: u crosses s.

3.4. Retrieving the reference movement pattern using PDP_S

In Section 3.2, we created a new dataset having 32 time points by transforming the original dataset from equal-interval-based to event-based. From here on, we refer to this new dataset as 'target dataset'. In this particular dataset, there is no change in the ordering among the vehicles along d_2, whereas the ordering does change along d_1. Therefore, for retrieving the reference movement pattern from the target dataset, we generated a set P consisting of all 3-tuples of the 16 vehicles in view of their d_1-precedence by scanning the entire target dataset.

As a next step, the d_1- and d_2-precedences of each tuple in P are compared with those of the reference movement pattern. Comparing both d_1- and d_2-precedences is important for finding an exact match; otherwise, tuples displaying a pass movement pattern where a vehicle is passing two vehicles on the other side in our reference movement pattern would also be detected as a match for the reference. If a tuple exactly follows the sequence of the reference movement pattern (i.e. five time points) for both descriptors along the period of 32 time points of the target dataset, then it is considered as an exact match of the reference movement pattern irrespective of the temporal length between the events. There might be an event-based matching, *i.e.*, a tuple of the target dataset matches with the precedences given at the first time point of the reference movement pattern from t_3–t_7, the second time point from t_8–t_{10} and the third time point from t_{11}–t_{17}, *etc*., or an equal-interval-based matching where a tuple matches with the reference movement pattern from t_3–t_7 or t_{24}–t_{28}. A tuple is dropped for further comparison if either there is a change in its d_1- or d_2-precedence as compared with that of the reference movement pattern or it follows the sequence of the reference movement pattern in a disordered way, e.g., it matches with the precedences at the second time point of the reference movement pattern from t_2–t_4, but it also matches with the precedences at the first time point of the reference movement pattern from t_5–t_8.

The above procedure is described in Table 2, where the pre-processing of the original dataset T takes place in lines 2–9, generating a new dataset T^{new} containing the time points at which a change occurs in the d_1-precedence or d_2-precedence of the vehicles as well as the precedences of the vehicles. Next, a set P containing all 3-tuples of the target vehicles O^T is generated in lines 10–13 for further analysis. The goal is to retrieve a list of tuples L matching with the d_1- and d_2-precedences of the reference objects (O^R) at time points t'_v, as implemented in lines 14–35. Note that if a tuple matches with the precedences at t'_2 of the reference dataset R first and follows the sequence of the reference movement pattern later on, it is also added to L.

Consider, for instance, that we need to verify whether tuple (o, h, g) is an exact match for the reference (u, t, s). Starting from t_1 of the target dataset, the d_1-precedence of (o, h, g), *i.e.*, $o < h < g$, and the d_2-precedence of (o, h, g), *i.e.*, $o < g = h$, are compared with

Table 2. Finding a reference movement pattern in a 2-D dataset using PDP_S.

Step	Procedure
1	Input a target dataset T of n point objects $O^T = \{o_1^T, o_2^T, \ldots, o_n^T\}$ whose location is tracked for u time points $t_1, t_2, \ldots, t_u$
2	Input a reference dataset R of r point objects $O^R = \{o_1^R, o_2^R, \ldots, o_r^R\}$ whose location is tracked for v time points $t'_1, t'_2, \ldots, t'_v$
3	Initialize L: A list of r-tuples from O^T matching the reference dataset R
4	$p_1 \leftarrow$ (d_1-precedence of O^T at t_1 of T, d_2-precedence of O^T at t_1 of T)
5	$T^{new} \leftarrow \{(t_1, p_1)\}$; //initialize a new dataset
6	for every two successive time points t_i, t_{i+1} of T do
7	$p_i \leftarrow$ (d_1-precedence of O^T at t_i, d_2-precedence of O^T at t_i)
8	$p_{i+1} \leftarrow$ (d_1-precedence of O^T at t_{i+1}, d_2-precedence of O^T at t_{i+1})
9	if $p_i \neq p_{i+1}$ //if the precedences are not equal
10	add (t_{i+1}, p_{i+1}) to T^{new}
11	$P \leftarrow \emptyset$
12	for each (t_i, p_i) in T^{new} do
13	$P \leftarrow P \cup$ {r-tuples of O^T at t_i according to p_i}
14	$L \leftarrow ()$
15	$w \leftarrow T^{new}$.size() //time points in the target dataset
16	$i \leftarrow 1$
17	for each tuple k in P do
18	$M \leftarrow ()$
19	$j \leftarrow 1$
20	while $i \leq w$ do
21	$q_i =$ (d_1-precedence of k at t_i, d_2-precedence of k at t_i)
22	while $j \leq v$ do //time points in the reference dataset
23	q_j= (d_1-precedence of O^R at t'_j, d_2-precedence of O^R at t'_j)
24	if $q_i = q_j$
25	add t'_j to M
26	$j = j + 1$
27	$i = i + 1$
28	if M. sort() = True
29	add k to L
30	return L

the d_1-precedence $u < t < s$ and the d_2-precedence $u < s = t$ of (u, t, s) at t'_1. The tuple (o, h, g) matches with the precedences at t'_1 of (u, t, s) from t_2–t_6. At t_7, (o, h, g) matches with the precedences of (u, t, s) at t'_2. From t_8–t_{29}, (o, h, g) matches with the precedences of (u, t, s) at t'_3. At t_{30}, (o, h, g) matches with the precedences of (u, t, s) at t'_4 and, at t_{31}, (o, h, g) matches with the precedences of (u, t, s) at t'_5. Hence, (o, h, g) is an exact match for (u, t, s).

Similarly, the same process is for instance repeated for another tuple (o, i, h). At t_1, the d_2-precedence of (o, i, h) is $o < i = h$, whereas the d_2-precedence of (u, t, s) is $u < s = t$. Since the d_2-precedence of (o, i, h) is identical to that of (u, t, s), we can say that (o, i, h) matches (u, t, s) along d_2. However, at t_1, (o, i, h) has d_1-precedence $i < o < h$ and (u, t, s) has d_1-precedence $u < t < s$. Clearly, there is a difference in the d_1-precedence of (o, i, h) as compared with (u, t, s). Hence, (o, i, h) is not a match for (u, t, s) along d_1 and cannot be taken for further analysis.

Finally, the exact matches for (u, t, s), which followed the reference movement pattern along d_1 and d_2 in the same sequence, were found to be (o, h, g) and (m, f, e).

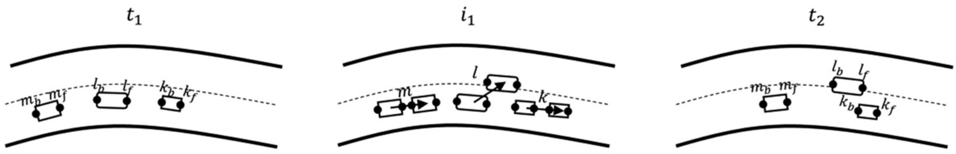
4. Extending PDP

So far, we have discussed the basics of PDP using simple micro-traffic examples and a real-world traffic application. We believe that certain extensions of the PDP representations might be closer to the real-world applications. We will now briefly explain these possible extensions one by one with some brief traffic examples.

4.1. Multiple points

Till now, we have only seen cases where PDP is explained by considering single points (using the centroid for each vehicle only). Depending on the context and the needs of the application, other points (such as front-end of the vehicle or multiple points per vehicle) can be represented as well. For instance, considering the front-end and back-end of the vehicle as two points, the PDP_S and PDP_D two-point representation for three vehicles (k, l and m) is shown in Figure 8. This representation is useful when the lengths of two vehicles (e.g., difference between vehicles and trucks) have to be taken into account.

Also, the centroid along with a safe following distance (which is much longer than a vehicle length) for each vehicle can be represented using PDP_S and PDP_D as shown in Figure 9. Considering a safe following distance into our representation, safe and dangerous events can be differentiated. For instance, by visualizing the safe following distance points between two vehicles, it can be seen which vehicle is maintaining a proper distance from the others and when there is a dangerous situation. The situations of sudden brake or improper lane-changing might be detected by adjusting the centroid and safety points of the rear vehicle with the back-end of the front vehicle. For example, a driving behaviour between vehicles m and k as shown in Figure 9(a) is considered safe if m_c and m_s remain in line with k_c and k_s along d_1 by maintaining the precedence as $m_c < m_s < k_c < k_s$. Nevertheless, the driving behaviour is considered as hasty or too close if the order between m_s and k_c reverses along d_1 as $m_c < k_c < m_s < k_s$. This means that the vehicle m needs to slow down and maintain an appropriate safe following distance from the vehicle k that is moving ahead of it.



(a)

$$\begin{cases} d_1: & m_b < m_f < l_b < l_f < k_b < k_f \\ d_2: & m_b = m_f = l_b = l_f = k_b = k_f \end{cases}$$

(b)

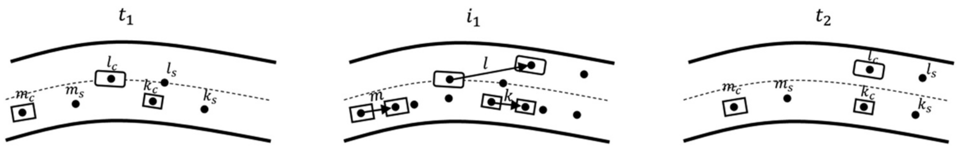
$$\begin{cases} d_1: & m_b^- < m_f^- < m_b^+ < m_f^+ < l_b^- < l_f^- < l_b^+ < k_b^- = l_f^+ < k_f^- < k_b^+ < k_f^+ \\ d_2: & l_b^+ = l_f^+ < m_b^- = m_f^- = m_b^+ = m_f^+ = l_b^- = l_f^- = k_b^- = k_f^- = k_b^+ = k_f^+ \end{cases}$$

(c)

$$\begin{cases} d_1: & m_b < m_f < l_b < l_f = k_b < k_f \\ d_2: & l_b = l_f < m_b = m_f = k_b = k_f \end{cases}$$

(d)

Figure 8. A motion scenario of three vehicles k, l and m for two time points t_1 and t_2 and during a time interval i_1 (a) with $\mathrm{PDP_S}$ ((b) and (d)), and $\mathrm{PDP_D}$ representations (c) for two-points ('f' for front-end and 'b' for back-end of each vehicle).



(a)

$$\begin{cases} d_1: & m_c < m_s < l_c < k_c < l_s < k_s \\ d_2: & l_c = l_s < m_c = m_s = k_c = k_s \end{cases}$$

(b)

$$\begin{cases} d_1: & m_c^- < m_c^+ < m_s^- < l_c^- < m_s^+ < k_c^- < l_s^- < l_c^+ = k_c^+ < k_s^- < l_s^+ = k_s^+ \\ d_2: & l_c^- = l_s^- = l_c^+ = l_s^+ < m_c^- = m_c^+ = m_s^- = m_s^+ = k_c^- = k_c^+ = k_s^- = k_s^+ \end{cases}$$

(c)

$$\begin{cases} d_1: & m_c < m_s < l_c = k_c < l_s = k_s \\ d_2: & l_c = l_s < m_c = m_s = k_c = k_s \end{cases}$$

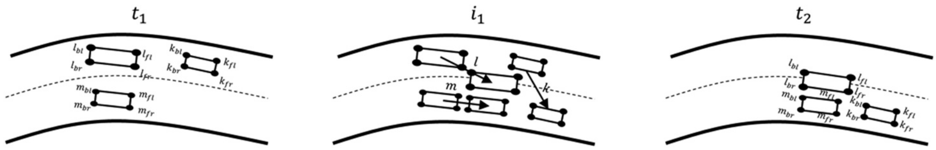
(d)

Figure 9. A motion scenario of three vehicles k, l and m (a) with $\mathrm{PDP_S}$ ((b) and (d)), and $\mathrm{PDP_D}$ representations (c) for two-points ('c' for centroid and 's' for safe following distance).

Another possibility is to consider the corner points of a vehicle and work with four points to take into account the width of vehicles. This is shown in Figure 10, where the two front points of a vehicle are represented by subscripts 'fl' and 'fr' and the rear points are represented by 'bl' and 'br'. The 'l' and 'r' represent the corresponding 'left' and 'right' points for the front and back sides of the vehicle. These points are important to detect, e.g., vehicles driving on a parking area or in a garage. The corner points of the garage might also be included for a deeper analysis of the situation.

PDP can be useful in differentiating identical movements taking place at different locations of the road by adding static and/or dynamic external points in its representation. The safe following distance point used in Figure 9 is actually an external dynamic point. To illustrate this, consider two identical movement patterns represented by three vehicles k, l and m at t_i and t_j respectively in Figure 11.

At t_i and t_j, the precedence of the centroids of the vehicles with respect to the static external point (b_c) on the bridge is given in Figure 11(b,c). By comparing the relational symbols along d_1, it is evident that the two identical movements actually account for a difference in locations and can be easily identified using PDP by adding the external point.



(a)

$$\begin{cases} d_1: & l_{bl} = l_{br} < m_{bl} = m_{br} < l_{fl} = l_{fr} = \\ & m_{fl} < m_{fr} < k_{bl} = k_{br} < k_{fl} = k_{fr} \\ d_2: & l_{bl} = l_{fl} = k_{bl} = k_{fl} < k_{br} = k_{fr} < \\ & l_{br} = l_{fr} < m_{bl} = m_{fl} < m_{br} = m_{fr} \end{cases}$$

(b)

$$\begin{cases} d_1: & l_{bl}^- = l_{br}^- < m_{bl}^- = m_{br}^- < l_{fl}^- = l_{fr}^- = m_{fl}^- = m_{fr}^- < \\ & l_{bl}^+ = l_{br}^+ = m_{bl}^+ = m_{br}^+ < m_{fl}^+ = m_{fr}^+ < k_{bl}^- = k_{br}^- = \\ & l_{fl}^+ = l_{fr}^+ < k_{bl}^+ = k_{br}^+ < k_{fl}^- = k_{fr}^- < k_{fl}^+ = k_{fr}^+ \\ d_2: & l_{bl}^- = l_{fl}^- = k_{bl}^- = k_{fl}^- < k_{br}^- = k_{fr}^- < l_{br}^- = l_{fr}^- < \\ & l_{bl}^+ = l_{fl}^+ < l_{br}^+ = l_{fr}^+ < m_{bl}^- = m_{fl}^- = m_{bl}^+ = m_{fl}^+ = \\ & k_{bl}^+ = k_{fl}^+ < k_{br}^+ = k_{fr}^+ < m_{br}^- = m_{fr}^- = m_{br}^+ = m_{fr}^+ \end{cases}$$

(c)

$$\begin{cases} d_1: & l_{bl} = l_{br} = m_{bl} = m_{br} < m_{fl} = m_{fr} < \\ & l_{fl} = l_{fr} < k_{bl} = k_{br} < k_{fl} = k_{fr} \\ d_2: & l_{bl} = l_{fl} < l_{br} = l_{fr} < m_{bl} = m_{fl} = \\ & k_{bl} = k_{fl} < k_{br} = k_{fr} < m_{br} = m_{fr} \end{cases}$$

(d)

Figure 10. A motion scenario of three vehicles k, l and m (a), with PDP_S (b) and (d), and PDP_D four-point representations (c).

(a)

$$\begin{cases} d_1: & k_c < l_c < m_c < b_c \\ d_2: & k_c < b_c < l_c = m_c \end{cases}$$

(b)

$$\begin{cases} d_1: & b_c < k_c < l_c < m_c \\ d_2: & k_c < b_c < l_c = m_c \end{cases}$$

(c)

Figure 11. Two identical movements at t_i and t_j (a), PDP_S representation using an external point for t_i and (b) t_j (c).

4.2. Multiple descriptors

PDP can take multiple distance descriptors into account for easy referencing depending upon the context of the application. Working with multiple descriptors might become useful to differentiate between different movement patterns. Consider for instance a traffic scenario where vehicle k is changing its lane from the right lane to the left lane (this is depicted as a bottom-to-top movement in Figure 12). The initial position of k with respect to its centroid during a certain interval (i_e or i_f) is represented as k_c^- whereas the final position is shown as k_c^+. It is evident that for this particular example, the precedence between the initial and final position of k is the same for the two descriptors d_1 and d_2.

By introducing a third descriptor (d_3), the change between the initial and final position of k becomes visible. If k_c^+ is quite far away from the coming vehicle m, this implies a steady and safe lane-changing event. If k_c^+ is quite close to the coming vehicle m, this implies a sudden lane-changing, hence can be termed as a dangerous event. In a nutshell, we can easily differentiate between safe lane-changing (Figure 12(a)) and dangerous lane-changing events (Figure 12(b)) with the aid of a third distance descriptor.

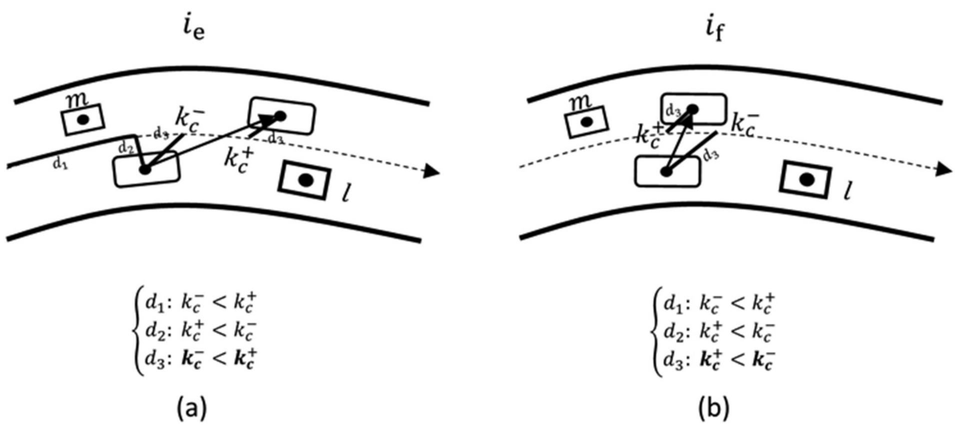


Figure 12. Safe lane-changing event (a) and dangerous lane-changing event (b).

Hence, having multiple descriptors in PDP may detect dangerous behaviour of vehicles at bottlenecks such as tunnels, bridges, cross-overs, roundabouts, or different locations on the road.

4.3. Extended PDP representations

The main advantage of PDP is its ability to represent information about movements and a possible extension of representing this information could be through an *Order Neighbourhood Diagram* (OND). The OND is an intuitive visualization of PDP, which depicts qualitative changes in the motion of the points. Unlike the traffic cellular automata models presented by Maerivoet and De Moor (2005), which describe the vehicles in different states of space-time diagrams, PDP could represent the vehicles in the corresponding nodes of the OND. The OND is a grid representation of nodes and edges where the nodes represent the points and the edges represent the conceptual neighbours using the precedence between these points. Just like PDP_S and PDP_D, ONDs can also be static and dynamic. Consider, for instance, a lane-changing event in Figures 13 and 14 with the corresponding static and dynamic ONDs shown in Figures 13(c) and 14 (c) respectively.

As is evident from Figures 13(c) and 14(c), the movement patterns of vehicles are tracked periodically and there is a 90° counter-clockwise rotational shift in the ONDs as compared to the actual motion scenario depicted in Figures 13(a) and 14(a). This is because we generally represent traffic in an upward direction in the ONDs. Note that the information required to generate ONDs is only dependent on the precedence of points. Moreover, the aspect of lane-changing is considered by incorporating an extra row (central line) in the OND.

For the purpose of simplicity, we make the following assumptions:

- each OND represents a 2-D grid of evenly spaced vertical and horizontal lines known as edges. The locations of points are represented as nodes on these edges at a specific time point or during a time interval,

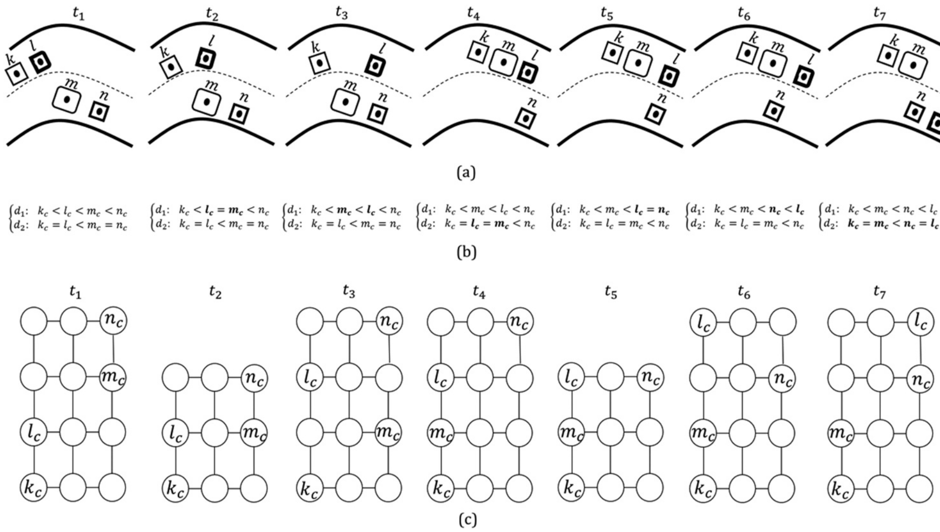


Figure 13. A lane-changing event (a), the precedence of vehicles described by each distance descriptor (b) and PDP$_S$-ONDs (c).

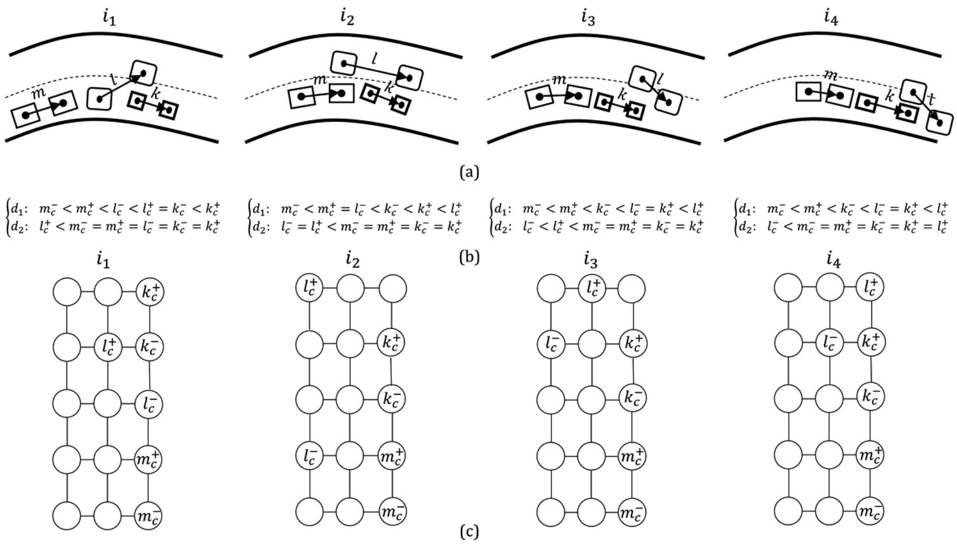


Figure 14. An overtake movement pattern (a) with the precedence of vehicles described by the two distance descriptors d_1 and d_2 (b). PDP$_D$-ONDs for each interval (c).

- no edge or node remains isolated or unconnected,
- each point holds a particular position on the nodes of the OND; no intermediate position is available across the edges,
- the length of a path followed by a moving point through OND is defined as the sum of the lengths of the edges along that path.

An important difference to highlight here is that the movement of vehicles along the road network is represented by moving between rows (horizontal lines) and columns (vertical lines) of the OND (Figure 13(c)). If a vehicle changes its lane, it moves across the columns, and if the precedence of vehicle changes, the movement is depicted across the respective rows of the OND. Note that a longer period might be represented via a sequence of ONDs.

5. Summary and future directions

Qualitative representation is an important area of artificial intelligence. So far, various qualitative representations have been developed, suited for a particular task. As an addition to the wide variety of qualitative representations, we have proposed a new qualitative approach (PDP) for representing points along a curved directed line through a system of relational symbols using different distance descriptors.

PDP uses the commonly available distance information about moving objects to represent their static and dynamic attributes. The overall goal of PDP is to facilitate the handling of the complexity of topological relations using a system of relational symbols. The PDP representations might be used for cluster analysis, top-*k* analysis, and calculating distance matrices. In addition, PDP can be potentially used for identifying events of particular interest from a certain motion dataset, as we illustrated in Section 3. For computational ease and analysis purposes, we discretized continuous time and performed our micro-traffic-analyses all at a temporal resolution of 10 Hz (Section 3). This resolution was chosen as optimal for the traffic examples after performing a rigorous analysis on a self-generated dataset of 26 variants of an overtake movement pattern on a two-lane road network. However, working with different resolutions might increase the applicability of PDP for specific applications such as finding a lane-changing event in a traffic scenario.

In Section 3, an approach for detecting a particular movement pattern in a given motion dataset has been discussed. Another alternative that might speed up the process of detection is to split the motion data into chunks (clusters) for analysis purpose rather than analyzing the entire data at once, as was suggested by Laube and Imfeld (2002) for the REMO-concept. For example, instead of taking all vehicles in the target dataset to generate tuples, we may apply a moving window around a vehicle that takes into account only the closest neighbours of that vehicle for generating tuples at a given moment (e.g., at t_1, take only 10 vehicles that are closest to vehicle *a* for generating tuples). In this way, the motion data can be analyzed systematically and the patterns might become more visible when objects are grouped based on their absolute and relative motion. Moreover, objects that are relatively far away and do not play a significant role in creating or analyzing events are excluded to avoid huge calculations.

Furthermore, we have discussed possible extensions of PDP through simple traffic motion patterns on a two- and three-lane road network. Besides, an additional visualization of PDP using OND has been discussed. The OND visualizations might look complex initially when dealing with a large number of vehicles/points; still, they are fast to interpret and comprehend.

It is pertinent to mention that the movement of interest can be a straight line, a circle (roundabout), or a spline (curved route) in PDP. The addition of extra information such as multiple distance descriptors and interactions with static/external points in PDP can potentially help seeking more insight into a practical phenomenon in a qualitative manner.

Another important aspect affecting the performance of PDP is the position of moving point objects in real-world scenarios. A small variation in the positions of these point objects might affect the ONDs. Similarly, in PDP we have to define the precedence of points as a function of certain threshold settings to minimize the effect of noise.

In summary, this study proposed a new qualitative approach to represent moving point objects in the spatiotemporal domain. Our approach is not only capable of representing curved line motion trajectories but can also discern between different moving patterns using the relative precedence between the points through multiple distance descriptors. This underpins the potential of representing a wide domain of movements with certain spatial constraints.

Furthermore, this paper only covers the basics of PDP by using simple micro-traffic examples. In the future, we plan to extend PDP to detect subtle motion patterns automatically, which could be useful in the context of traffic safety. Moreover, it would be more interesting to add contextual information in PDP. By analysing every possible spatial arrangement of points (permutation) in PDP, the similarity analysis will be independent of the points' configuration. This might lead to a well-calibrated pattern matching tool that could be beneficial for detecting or searching a particular motion pattern from a motion database in the fields related to autonomous vehicles.

The applicability of PDP will be extended to other domains of constrained mobility like sports and gait analysis where the movement trajectories vary in length. Hence, a possible extension could be applying PDP to complex movement patterns having different lengths using the Levenshtein technique (Beernaerts *et al.* 2018). Also, for the successful application of PDP in artificial intelligence applications such as human-robot interaction, it is interesting to investigate how PDP could take a qualitative description as input and return the respective trajectory in an inverse way. This inverse problem can certainly be handled using PDP_D and could be a rich basis for studying point configurations and movements.

Acknowledgments

The authors would like to acknowledge Dilawar Ali for his valuable assistance in image processing and the creation of the dataset related to this paper.

Data and codes availability statement

The data and codes that support the findings of this study are available with a DOI at https://doi.org/10.6084/m9.figshare.12248318.v2.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Higher Education Commission (HEC), Pakistan [50040696].

Notes on contributors

Amna Qayyum is a Ph.D. research scholar at the CartoGIS research unit of Ghent University. Her research focuses on automated vehicle analysis using qualitative spatiotemporal techniques at a microscale level.

Bernard De Baets holds a Ph.D. degree in Mathematics. He is currently a full Professor at Ghent University, where he is leading the research unit Knowledge-Based Systems (KERMIT) as well as Department of Mathematical Modelling, Statistics and Bio-informatics. He is also a Co-Editor-in-Chief of Fuzzy Sets, Systems, and a member of the Editorial Board of several other journals. He has delivered over 200 (invited) conference lectures.

Muhammad Sulman Baig is a Ph.D. research scholar at the CartoGIS research unit of Ghent University. His work deals with the understanding of driving behaviour of human drivers that can potentially be used in developing a decision making tool for autonomous vehicles. His work also includes studying human behaviour and preferences towards the autonomous vehicle technology that could have effect on future policies concerning the autonomous technology.

Frank Witlox has a PhD in urban design (Eindhoven University of Technology, supervisor Harry Timmermans). He is currently the department chair and full professor of economic geography at the Department of Geography of Ghent University.

Guy De Tré is Associate professor at the Department of Telecommunications and Information Processing of the Faculty of Engineering and Architecture of Ghent University. He is head of the Database, Document, and Content Management (DDCM) research group. His research activities are centred on computational intelligence in information management systems, including fundamental research on bipolarity handling, uncertainty handling, multi-valued logics and spatio-temporal modelling and applied research on, among others, big data (NoSQL databases), fuzzy querying, decision support, data quality, and unstructured data.

Nico Van de Weghe is a geomatics professor at Ghent University. He is specialized in geographic information science (focus on the analysis and modelling of spatial-temporal information, ranging from mobility research to sports analysis) and has a broad experience in setting up practical experiments in the field of geographic information technology (focus on tracking moving objects and location-based services). is a geomatics professor at Ghent University. He is specialized in geographic information science (focus on the analysis and modelling of spatial-temporal information, ranging from mobility research to sports analysis) and has a broad experience in setting up practical experiments in the field of geographic information technology (focus on tracking moving objects and location-based services).

ORCID

Amna Qayyum http://orcid.org/0000-0003-0464-4308
Bernard De Baets http://orcid.org/0000-0002-3876-620X

References

Arrufi, J.P. and Kirsch, A., 2018. Motion categorisation: representing velocity qualitatively. *Cognitive Systems Research*, 52, 117–131. doi:10.1016/j.cogsys.2018.06.005

Balbiani, P. and Condotta, J.-F., 2002. Spatial reasoning about points in a multidimensional setting. *Applied Intelligence*, 17 (3), 221–238. doi:10.1023/A:1020079114666.

Beernaerts, J., *et al.*, 2018. A method based on the Levenshtein distance metric for the comparison of multiple movement patterns described by matrix sequences of different length. *Expert Systems with Applications*, 115, 373–385. doi:10.1016/j.eswa.2018.07.076.

Clementini, E., 2013. Directional relations and frames of reference. *GeoInformatica*, 17 (2), 235–255. doi:10.1007/s10707-011-0147-2.

Condotta, J.-F., *et al*., 2006. A generic toolkit for n-ary qualitative temporal and spatial calculi. *In*: *13th International Symposium on Temporal Representation and Reasoning*. Budapest: IEEE Computer Society, 78–86.

Delafontaine, M., *et al*., 2011. Inferring additional knowledge from QTCN relations. *Information Sciences*, 181 (9), 1573–1590. doi:10.1016/j.ins.2010.12.021.

Dylla, F. and Lee, J.H., 2010. A combined calculus on orientation with composition based on geometric properties. *In*: *ECAI 2010-19th European Conference on Artificial Intelligence*, 16–20 August. Lisbon, Portugal: IOS Press, 1087–1088.

Frank, A.U., 1996. Qualitative spatial reasoning: cardinal directions as an example. *International Journal of Geographical Information Systems*, 10 (3), 269–290. doi:10.1080/02693799608902079.

Glez-Cabrera, F.J., *et al*., 2013. QRPC: A new qualitative model for representing motion patterns. *Expert Systems with Applications*, 40, 4547–4561. doi:10.1016/j.eswa.2013.01.058.

Isli, A. and Cohn, A.G., 2000. A new approach to cyclic ordering of 2D orientations using ternary relation algebras. *Artificial Intelligence*, 122 (1–2), 137–187. doi:10.1016/S0004-3702(00)00044-8.

Kumar, D., 2018. Top view of traffic moving on a busy freeway. Downtown Los Angeles. Available from: https://www.youtube.com/watch?v=I2qEIr0Te2U [Accessed 5 Dec 2018].

Kurata, Y. and Shi, H., 2008. Interpreting motion expressions in route instructions using two projection-based spatial models. *In*: A.R. Dengel, *et al*., eds. *Advances in artificial intelligence*, 23–26 September. Kaiserslautern, Germany: Springer, 258–266.

Laube, P. and Imfeld, S., 2002. Analyzing relative motion within groups of trackable moving point objects. *In*: M.J. Egenhofer and D.M. Mark, eds. *Geographic information science. Lecture notes in computer science*. Heidelberg, Berlin: Springer, 132–144.

Li, S. and Liu, W., 2015. Cardinal directions: a comparison of direction relation matrix and objects interaction matrix. *International Journal of Geographical Information Science*, 29 (2), 194–216. doi:10.1080/13658816.2014.954580.

Maerivoet, S. and De Moor, B., 2005. Cellular automata models of road traffic. *Physics Reports*, 419 (1), 1–64. doi:10.1016/j.physrep.2005.08.005.

Moratz, R., 2006. Representing relative direction as a binary relation of oriented points. *In*: *ECAI 2006, 17th European Conference on Artificial Intelligence*, 29 August-1 September.Riva del Garda, Italy: IOS Press, 407–411.

Mossakowski, T. and Moratz, R., 2012. Qualitative reasoning about relative direction of oriented points. *Artificial Intelligence*, 180–181, 34–45. doi:10.1016/j.artint.2011.10.003

Peuquet, D.J. and Duan, N., 1995. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Systems*, 9 (1), 7–24. doi:10.1080/02693799508902022.

Skiadopoulos, S. and Koubarakis, M., 2001. Composing cardinal direction relations. *Artificial Intelligence*, 152 (2), 143–171. doi:10.1016/S0004-3702(03)00137-1.

Skiadopoulos, S. and Koubarakis, M., 2005. On the consistency of cardinal direction constraints. *Artificial Intelligence*, 163 (1), 91–135. doi:10.1016/j.artint.2004.10.010.

Van de Weghe, N., *et al*., 2005. Representing moving objects in computer-based expert systems: the overtake event example. *Expert Systems with Applications*, 29 (4), 977–983. doi:10.1016/j.eswa.2005.06.022.

Vilain, M. and Kautz, H., 1986. Constraint propagation algorithms for temporal reasoning. *In*: *5th AAAI National Conference on Artificial Intelligence*, 11–15 August.Philadelphia, Pennsylvania: AAAI Press, 377–382.

Zimmermann, K. and Freksa, C., 1996. Qualitative spatial reasoning using orientation, distance, and path knowledge. *Applied Intelligence*, 6 (1), 49–58. doi:10.1007/BF00117601.