Anchor uncertainty and space-time prisms on road networks

Bart Kuijpers, Harvey J. Miller, Tijs Neutens & Walied Othman

To cite this article: Bart Kuijpers, Harvey J. Miller, Tijs Neutens & Walied Othman (2010) Anchor uncertainty and space-time prisms on road networks, International Journal of Geographical Information Science, 24:8, 1223-1248, DOI: 10.1080/13658810903321339

To link to this article: https://doi.org/10.1080/13658810903321339

Published online: 16 Jun 2010.

Submit your article to this journal

Article views: 493

View related articles

Citing articles: 46 View citing articles
Anchor uncertainty and space-time prisms on road networks

Bart Kuipersa, Harvey J. Millerb, Tijs Neutensc and Walied Othmana*

aTheoretical Computer Science Group, Hasselt University, Agoralaan, Gebouw D, 3590 Diepenbeek, Belgium; bDepartment of Geography, 260 S. Central Campus, Dr. Room 270, University of Utah, Salt Lake City, UT 84112-9155, U.S.A; cDepartment of Geography, Ghent University, Sint-Pietersnieuwstraat 25, B–9000 Ghent, Belgium

(Received 12 January 2009; final version received 4 September 2009)

Space-time prisms capture all possible locations of a moving person or object between two known locations and times given the maximum travel velocities in the environment. These known locations or ‘anchor points’ can represent observed locations or mandatory locations because of scheduling constraints. The classic space-time prism as well as more recent analytical and computational versions in planar space and networks assume that these anchor points are perfectly known or fixed. In reality, observations of anchor points can have error, or the scheduling constraints may have some degree of pliability. This article generalizes the concept of anchor points to anchor regions: these are bounded, possibly disconnected, subsets of space-time containing all possible locations for the anchor points, with each location labelled with an anchor probability. We develop two algorithms for calculating network-based space-time prisms based on these probabilistic anchor regions. The first algorithm calculates the envelope of all space-time prisms having an anchor point within a particular anchor region. The second algorithm calculates, for any space-time point, the probability that a space-time prism with given anchor regions contains that particular point. Both algorithms are implemented in Mathematica to visualize travel possibilities in case the anchor points of a space-time prism are uncertain. We also discuss the complexity of the procedures, their use in analysing uncertainty or flexibility in network-based prisms and future research directions.

Keywords: moving objects; uncertainty; space-time prisons; road networks; GIS; movement modeling; quantifying uncertainty

1. Introduction

The space-time prism (Hägerstrand 1970) is a powerful method for analysing potential human activity in an environment. The prism demarcates all locations in space where a person can be during a time window given the achievable travel velocities and the locations and times where the person or object must be at the beginning and end of the time window. Researchers have developed analytical and computational tools for calculating prisms with planar space and transportation networks (Miller 1991, 2005a, b, Kwan and Hong 1998, Yu and Shaw 2008). Prisms can measure an individual’s accessibility to an environment as well as uncertainty about the location of moving objects between locational fixes. Intersections between prisms show the potential for human or object contact (Neutens et al. 2007b). Their strong theoretical foundation combined with increasingly powerful
tools and available data means that space-time prisms are being applied in a wide-range of contexts, including mobile objects databases (Kuijpers and Othman 2009), transportation planning (Timmermans et al. 2002), epidemiology and environmental risk assessment (Löytönen 1998, Jacquez 2000, Jacquez et al. 2005), social research (Kwan 1999) and crime analysis (Ratcliffe 2006), just to name a few domains.

A limitation of classical space-time prisms is the unrealistic assumption that the start and end points of an individual’s time window are perfectly fixed or known in both space and time; these points are often referred to as the prism anchors. In practice, prism anchors often have a substantial degree of measurement error or inherent flexibility. For example, location-aware technologies (LATs) (such as the global positioning system (GPS)) often have error. The self-reported activity data provided by individuals (often in retrospective) will contain error or uncertainty about the exact location and timing of events. Finally, it is often the case that anchors are inherently flexible: mandatory activities such as meetings or work can nevertheless have a degree of pliability with respect to their timing and even locations. Anchor uncertainty or flexibility propagates to the prism itself in ways that are unknown and not captured by current planar or network prism methods.

This article presents a generalization of the network-based space-time prism that allows prism anchors to be uncertain or flexible with respect to space and time. Rather than fixed points in space and time, prism anchors are a finite set of possible outcomes, each with a degree of uncertainty. The resulting anchor regions propagate to the network time prism, with the result that locations in the prism have degrees of uncertainty. We develop two algorithms for calculating network-based space-time prisms based on these probabilistic anchor regions. The first algorithm calculates the envelope of all space-time prisms having an anchor point within a particular anchor region. The second algorithm calculates, for any space-time point, the probability that a space-time prism with given anchor regions contains that particular point. Both algorithms have reasonable worst-case complexity. We illustrate the theory and methods through some example calculations using the Mathematica software. We also discuss the use of these procedures in analysing network prism uncertainty and flexibility based on anchor regions.

The remainder of the article is organized as follows: We begin with a brief overview of the relevant literature in Section 2. In Section 3, we start with defining space-time prisms, trajectories, road networks and trajectories and space-time prisms on road networks. Section 4 then introduces the notion of anchor regions and uncertain space-time prisms. In this section we also present an algorithm to compute and visualize the uncertain space-time prism. Section 5 introduces the algorithm to compute the fraction of the uncertain prism that covers a given spatio-temporal point and the probability of that fraction. Moreover, we implemented this in Mathematica to visualize uncertain space-time prisms and the probability in each spatio-temporal point of this prism. In Section 6, we discuss the complexity issues of the algorithm. In Section 7, we illustrate some applications of our results. Finally, we conclude with the major findings and outline the avenues for future research.

2. Relevant literature

One of the dominant ideas underlying the activity-based modelling approach to travel forecasting is that while interacting and performing activities, individuals are faced with the inseparability and scarce nature of space and time. Individual movement implies a trade-off between these resources and is conditioned by the various constraints and opportunities offered by the urban, physical and institutional context in which individuals are embedded (Timmermans et al. 2002, Miller 2007). These ideas were already recognized as early as the late 1950s in the seminal work by Hägerstrand (1957, 1970) and
became known as *time geography*. A key concept of time geography is the *space-time prism*, which captures potential human movement between two discrete space-time points, also often referred to as anchor points. Classical space-time prisms assume a uniform travel velocity in an isotropic and homogeneous space. Therefore, they tend to overestimate individual travel possibilities, given that individuals are usually confined to transportation networks with link-specific travel velocities. Hence, they have merely conceptual value but are inadequate for analytically solving problems concerning the extent of individual access. A number of scholars have addressed this issue and proposed geocomputational algorithms to implement space-time prisms within networks relying on shortest path calculations (Miller 1991, Kwan and Hong 1998, Wu and Miller 2002, Kuipers and Othman 2009), isochrones (O’Sullivan *et al.* 2000, Neutens *et al.* 2008a) and velocity fields (Miller and Bridwell 2009).

These network-based implementations of space-time prisms assume that the spatio-temporal coordinates of the anchor points are exactly known and strictly fixed in space as well as in time. In reality, however, this is typically not the case as uncertainty may arise from different sources related to the ways in which anchor points of space-time prisms are commonly defined or measured. Firstly, in research on individual space-time accessibility, it has become common practice to define anchor points as the start and end points of fixed activities and to derive these retrospectively from activity and travel diary data (Weber and Kwan 2002, Kim and Kwan 2003a). As noted by Rietveld (2002) and Witlox (2007), the accuracy of reported departure and arrival points is affected by rounding and geocoding imprecision that might bias the assessment of individual accessibility. Secondly, individual movement can be recorded by means of discrete time-stamped anchor points obtained with high resolution using-LATs such as the GPS and radio-location methods (Hammand and Karimi 2004, Miller 2005b). The use of such devices implies both measurement and sampling errors. Although sampling errors relate to the observation frequency with which the anchor points are recorded, measurement errors result from the inaccuracy inherent to the tracking technique used. In the case of GPS, measurement errors are often modelled as a bivariate normal distribution in the \((x, y)\)-plane (Pföser *et al.* 2005). Third, and perhaps, most problematic for representing an individual’s travel possibilities is the uncertainty on the part of the individuals themselves. When making scheduling decisions, individuals have to reckon with the uncertain performance of the transportation system and the possibly uncertain duration and location of their future space-time requirements (Hendricks *et al.* 2003). Finally, as argued by Schwanen (2006), considering anchor points as temporally fixed is questionable as arriving on time may in many cases be conceived of as a range of acceptable arrival times rather than a single clock time.

Despite the variety of sources from which uncertainty in anchor points may arise, until now the main focus has been directed towards the management of uncertainty in moving object databases (MOD) (Pföser and Jensen 1999, Hornsby and Egenhofer 2002, Wolfson 2002, Trajcevski *et al.* 2004), handling indeterminate locations in spatial information systems (Hood and Galton 2006) and the effects of uncertain travel times because of systematically recurring congestion or unreliable transportation systems (Hall 1983, Ettema and Timmermans *et al.* 2002, 2007). The equally important question as to what extent uncertain anchor points might influence the possibilities for travel and activity participation, however, has received only scant attention in literature on individual space-time accessibility analysis. This issue pertains to the degree of flexibility of the temporal boundaries of activities. Notable exceptions include Hendricks *et al.* (2003) and Neutens *et al.* (2007a), both relying on classical time geography and consequently assuming an isotropic and homogeneous travel environment. Another exception are Schwanen and de Jong (2008), who incorporated and represented the flexibility in the duration of time windows in their case study of space-time accessibility.
3. Preliminaries

In this section, we outline some basic definitions that are necessary for the development of our algorithms in the remainder of the article.

Definition 3.1: Let $\mathbb{R}$ denote the set of real numbers. A road network $\mathbb{R}N$ is a graph embedding in $\mathbb{R}^2$ of a labelled graph given by a finite set of vertices $V = \{(x_i, y_i) \in \mathbb{R}^2 | i = 1, \ldots, N\}$ and a set of edges $E \subseteq V \times V$ that are labelled by a speed limit and an associated time span. This graph embedding satisfies the following conditions: Edges are embedded as straight line segments between vertices. If an edge between $(x_i, y_i)$ and $(x_j, y_j)$ is labelled by the speed limit $v_{ij} > 0$, then its time span $w_{ij}$ is $\frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{v_{ij}}$, i.e., it is the time needed to get from one side of an edge to another when travelling at the speed limit.

This yields $\mathbb{R}N = \{(x, y) = (1 - \lambda)(x_i, y_i) + \lambda(x_j, y_j) | \lambda \in [0, 1] \text{ and } ((x_i, y_i), (x_j, y_j)) \in E\} \cup V$.

A trajectory can be anything continuous, be it a polyline or something more differentiable. It is assumed that trajectories are recorded as a discrete list of time-stamped locations, called anchor points, and that nothing is known about an object’s position between those anchor points other than an upper bound on its speed. Moreover, we assume all our trajectories to be part of the road network.

Definition 3.2: First, we define trajectories and then their restriction to a road network.

- Let $I \subseteq \mathbb{R}$ be an interval. A trajectory $T$ is the graph of a mapping $\alpha : I \rightarrow \mathbb{R}^2 : t \mapsto \alpha(t) = (\alpha_x(t), \alpha_y(t))$, i.e., $T = \{(t, \alpha_x(t), \alpha_y(t)) \in \mathbb{R} \times \mathbb{R}^2 | t \in I\}$. We call $I$ the time domain of $T$.
- A trajectory sample is a finite set $S = \{(t_0, x_0, y_0), (t_1, x_1, y_1), \ldots, (t_N, x_N, y_N)\}$ of space-time points. The order on time, $t_0 < t_1 < \cdots < t_N$, induces a natural order on the sample.
- If $T$ is a trajectory given by the functions $\alpha_x$ and $\alpha_y$, then it must satisfy $(\alpha_x(t), \alpha_y(t)) \in \mathbb{R}N$ for all $t$ in the time domain of $T$ and for a trajectory sample $S = \{(t_0, x_0, y_0), (t_1, x_1, y_1), \ldots, (t_N, x_N, y_N)\}$ we must have $(x_i, y_i) \in \mathbb{R}N$ for all $i = 0, \ldots, N$. A trajectory (sample) on a $\mathbb{R}N$ is a trajectory (sample) whose spatial projection is in $\mathbb{R}N$, as illustrated in Figure 1.

Figure 1 shows a road network and a trajectory, as well as a trajectory sample, on top of it.

We now define space-time prisms. In the traditional sense, a space-time prism is the union of all trajectories from one point to another that are constraint by a speed limit. A space-time prism for an object moving in the real plane, unconstrained in its movement, is defined as follows (Hägerstrand 1970, Egenhofer 2003, Miller 2005b):

Definition 3.3: Let $p = (x_p, y_p), q = (x_q, y_q) \in \mathbb{R}^2$. The set of points $(t, x, y)$ in the space-time prism with origin $(t_p, p)$, destination $(t_q, q)$ and maximal speed $v_{\text{max}} > 0$ satisfy the constraints

\[
\begin{align*}
(x - x_p)^2 + (y - y_p)^2 & \leq (t - t_p)^2 v_{\text{max}}^2 \\
(x - x_q)^2 + (y - y_q)^2 & \leq (t_q - t)^2 v_{\text{max}}^2 \\
t_p \leq t & \leq t_q
\end{align*}
\]
This space-time prism is visualized in Figure 2 and denoted by $\mathcal{P}^{\mathbb{R}^2}(t_p, x_p, y_p, t_q, x_q, y_q, v_{\text{max}})$.

Space-time prisms can be adapted to road networks with either uniform speed limits over the entire network or speed limits that may vary per edge, and for bi- or unidirectional edges.

To define space-time prisms on a road network, an appropriate distance function on the network is necessary. This distance measure that is used here is derived from the shortest path-distance used in graph theory (Weisstein 2007).

**Definition 3.4:** Let $\mathbb{R}^n$ be a road network, $p, q \in \mathbb{R}^n$ and let $V_{pq}$ be the set of vertices $V \cup \{p, q\}$ and let $E_{pq}$ equal the set of edges obtained from $E$ by adding $p$ and $q$ to the vertex set. The *road network time* between $p$ and $q$, denoted by $d_{\mathbb{R}^n}(p, q)$, is the shortest-path
distance between \(p\) and \(q\) in the graph \((V_{pq}, E_{pq})\), with respect to the time-span labelling of the edges.

A path from \(p\) to \(q\) whose length, with respect to the time-span labelling of the edges, equals \(d_{RN}(p, q)\) is called a fastest path from \(p\) to \(q\).

Note that the road network time between \(p\) and \(q\) in the above definition has minimal total weight and equals the shortest time span in which you can reach \(q\) from \(p\). The metric takes two points from a road network as input and returns the shortest time needed to get from one to the other when travelling at the allowed maximal speed at each segment.

A space-time prism on a road network is the geometric location in \(\mathbb{R} \times RN \subset \mathbb{R} \times \mathbb{R}^2\) of all points a moving object could have visited when travelling, restricted to \(RN\), from an origin \(p\) to a destination \(q\) within a time-frame ranging from \(t_p\) to \(t_q\), respecting the speed limits on the edges of \(RN\). Given a road network \(RN\), points \(p, q\) and \(u\) on \(RN\), and time moments \(t_p\) and \(t_q\) for \(p\) and \(q\), we write \(t_u^-\) to abbreviate \(t_p + d_{RN}(p, u)\) and \(t_u^+\) to abbreviate \(t_q - d_{RN}(u, q)\).

**Definition 3.5:** Let \(RN\) be a road network, let \(p, q \in RN\). The space-time prism on the road network between \((t_p, p)\) and \((t_q, q)\), with respect to the speed limits of \(RN\), is denoted by \(P^{RN}(t_p, p, t_q, q)\) and is defined as the set of \((t, u)\) tuples, where \(u \in \mathbb{R}^2\) for which

\[
\begin{align*}
u & \in RN, \\
d_{RN}(p, u) + d_{RN}(u, q) & \leq (t_q - t_p), \\
t_u^- & \leq t \leq t_u^+.
\end{align*}
\]

The space-time prism on the road network between \((t_p, p)\) and \((t_q, q)\), with respect to a general maximal speed \(v_{\text{max}}\), is the space-time prism on \(RN\) after relabelling all edges of \(RN\) with speed limit \(v_{\text{max}}\). 

Figure 3 represents a space-time prism on a road network and was generated using a previously implemented algorithm in Mathematica, which is available in Othman (2007).
4. From uncertainty on anchor points to anchor regions and uncertain space-time prisms

In the preliminaries, the anchor points are considered to be perfectly known and are represented by space-time points. As argued earlier, in real life, a lot of sources may introduce uncertainty about anchor points. Measurement errors, for one, are introduced when measuring locations using GPS, for example. When a person is asked to keep track of his/her locations and time spent there, errors get easily introduced, for example, ‘I left work between 5 and 5:30 pm’. To cope with these errors, we extend the concept of certain anchor points to anchor regions.

4.1. Uncertain anchor points

For simplicity, we start by defining an anchor region on a line segment and later expand this definition to road networks. A simplification in our model is that time and space are considered independent from each other. This ensures that anchor regions are box-shaped in space-time and that our model behaves in a polygonal fashion for tractability. This box-shaped region will be termed an anchor region. Independence between space and time is a reasonable approximation of many real-world situations. Although some positioning methods in LATs use signal arrival times, time at the client to fix location (e.g., the GPS), other methods use the angles or strengths of arriving signals as well as triangulation and dead-reckoning, where spatial and temporal errors are independent (Roth 2004). Also, positional error in LAT tends to be dominated by finite spatial sampling rather than measurement error in the locational fixes and times (Pfoser and Jensen 1999). In self-reported data, a user’s uncertainty about the location of an event does not necessarily imply uncertainty about its timing, and vice versa. Nevertheless, a more general model would allow error dependence between space and time. This is a topic for further investigation.

Moreover, in this anchor region some subsets can be more likely than others. This can be described using probability functions. We refer to Figure 4 for a conceptual representation of this model.

**Definition 4.1:** An anchor region on a segment is a bounded subset of space-time of all possible locations of an anchor point. Let \( p = (x_p, y_p), q = (x_q, y_q) \in \mathbb{R}^2 \) be the spatial borders and \( t_{pq}^-, t_{pq}^+ \in \mathbb{R} \), the temporal borders of the region, meaning the region is bounded by the polygon \(((t_{pq}^-, x_p, y_p), (t_{pq}^+, x_p, y_p), (t_{pq}^+, x_q, y_q), (t_{pq}^-, x_q, y_q))\). This region is encoded by the six-tuple \( S_{pq} = (p, q, t_{pq}^-, t_{pq}^+, \mu_{pq}, \nu_{pq}) \), where \( \mu_{pq} : [t_{pq}^-, t_{pq}^+] \rightarrow \mathbb{R}^+ \) and \( \nu_{pq} : [t_{pq}^-, t_{pq}^+] \rightarrow \mathbb{R}^+ \) are independent probability functions and \( \lambda_{pq} : \lambda \mapsto \lambda_{pq}(\lambda) \), where \( \lambda_{pq} \) is a probability function on the line \((x, y) = (1 - \lambda)p + \lambda q\).

Note that the definition above does not require that \( \lambda_{pq} \) is restricted to \([0,1]\). Indeed, \( \lambda_{pq} \) can be defined on the entire line supported by \( p \) and \( q \). However, on a road network \( \mathbb{R}^n \), we can model a moving object’s location using a finite number of, possibly disconnected, anchor regions. The probability functions, restricted to all these regions combined, have to integrate to one.

**Definition 4.2:** An anchor region \( S \) on a road network \( \mathbb{R}^n \) is a bounded subset of space-time of all possible locations of an anchor point on \( \mathbb{R}^n \). In particular, it is a finite set of anchor regions on segments as defined in Definition 4.1, \( S = \bigcup_i S_i \) where \( i = 1, \ldots, n \) and each \( S_i \) is an anchor region on a segment. Moreover, all the \( S_i \) are disjoint and integrating all the spatial probability functions over all the spatial components of these anchor regions adds up to one.
Remark 1

- Note that it is not required that the regions stitch nicely together in time. This allows modelling locations that are accessible at discrete intervals in time. For example, shopping malls that are not open 24 hours a day.
- Second, it is not required that they are continuous in space either. This, in turn, allows modelling multiple separated probable departure and arrival locations, and, as we will soon elaborate on, calculating a relative likelihood for each region.

4.2. Uncertain space-time prisms

The next step is to adapt the space-time prism model to these anchor regions. This can be done in a straightforward manner as described in the next definition.

Definition 4.3: Let $\mathcal{RN}$ be a road network and $S_b$ and $S_e$ the anchor regions on $\mathcal{RN}$. An uncertain space-time prism between the anchor regions $S_b$ and $S_e$ is the union of all space-time prisms with starting point in $S_b$ and ending point in $S_e$. An additional constraint is that there needs to exist a space-time prism from every point in $S_b$ that has an endpoint in $S_e$ and vice versa.

When every point in an anchor region is equally probable, then the probability function for time and space is a uniform distribution. To model those points that are more probable around the centre of an anchor region than at the edges, a normal distribution can be used.

4.3. Computing the envelope of the uncertain prism

In this section, an algorithm is introduced that computes the uncertain space-time prism, which envelopes the union of all space-time prisms that connect a point from the starting anchor regions to a point in the ending anchor regions. This algorithm is an adaptation of earlier work (Kuijpers and Othman 2009) where the proofs of the correctness of this algorithm can be found. In our algorithm, we consider each edge separately and it suffices to compute the earliest arrival and latest departure times for both vertices of that edge.
The following observations will simplify the computations significantly: For each edge, the uncertain prism is constructed, we cycle through all the edges of the starting anchor region. For each such edge, we note that the fastest path has to pass over one of the nodes of that edge. Moreover, a path from the interior of such an edge is longer than a path that departs from at least one of the nodes. So the earliest time you can reach a node is the minimum of the road network distances to that node from all nodes of the departure anchor region at the earliest time of that region. Likewise, the latest departure times at each node will equal the maximum of all road network distances to all nodes of the arrival anchor region at the latest time of that region.

The precomputation algorithm consists of two steps. In the first step we adapt the road network so that the spatial boundaries of the anchor regions become vertices of the road network. In the second step, the earliest arrival and latest departure times are computed for each node. These computation steps are not sufficient for the edges that support anchor regions, because both steps do not take into account the points in the interior of an edge.

\[ PRECOMP : \text{input} = (V, E, S_b, S_e); \]
\[ \text{output} = (RN', \{(t_u^-, t_u^+) \mid u \in RN'\}, V_P, E_P) \]

**Step 1:** In this step, vertices are added to the network.
- For all \( S_{b,i}, S_{e,j} \), let such an anchor region be spatially bounded by \( p, q \in \mathbb{R}^2 \). If \( p \) is a vertex then do nothing, else let \( r, s \in \mathbb{R}^2 \) be the vertices that bound the edge \([r, s]\) that contains \( p \). Remove that edge from the network, add the vertex \( p \) to the network and add the edges \([r, p]\) and \([p, s]\) to the network. Repeat the same procedure for \( q \).

**Step 2:** In this step, the earliest arrival time and latest departure time are computed in each vertex, with respect to all the anchor regions where an object can leave from and all regions where it can arrive at. Dijkstra's single source-shortest path algorithm is adapted. This modification consists of altering the stop-condition. Although in the original version, the algorithm stops when all vertices were processed, in our modified version, we stop when either all vertices are processed, or when the distance to the last processed vertex is greater than the distance the object is able to travel in the given time-frame.

Let \( t_{\text{max}} = \max_{ij} \{t_{e,j}^+ - t_{b,i}^- \mid S_{b,i} = (p_{b,i}, q_{b,i}, t_{b,i}^-, t_{b,i}^+, \mu_{b,i}, \chi_{b,i}) \text{ and } S_{e,j} = (p_{e,j}, q_{e,j}, t_{e,j}^-, t_{e,j}^+, \mu_{e,j}, \chi_{e,j}) \} \). Initialize all vertices \( u \) in the road network with \( t_u^- = -\infty \) and \( t_u^+ = +\infty \). The following steps need to be repeated for all \( \{r \mid \exists i : S_{b,i} = (p, q, t^-, t^+, \mu, \chi) \text{ or } \exists j : S_{e,j} = (p, q, t^-, t^+, \mu, \chi) \text{ and } r = p \text{ or } r = q \} \):

1. Let \( D = \{(s, t_s) \mid s \text{ be a vertex and } t_s = \infty \text{ if } s \neq r \text{ and } t_s = 0 \text{ otherwise}\} \).
2. Let \((s, t_s) \in D\) be the pair with the smallest value of \( t_s \). Store \( s \) in \( V_P \). Remove \((s, t_s)\) from \( D \).
3. If \( r \) belongs to a \( S_{b,i} = (p, q, t^-, t^+, \mu, \chi) \) then set \( t_s^- = \min\{t_e^-, t_e^- + t_e\} \), else if \( r \) belongs to a \( S_{e,j} = (p, q, t^-, t^+, \mu, \chi) \) then set \( t_s^+ = \max\{t_e^+, t_e^- + t_e\} \).
4. For all the edges \([s,u]\) connected to \( s \) where there exists a \( t_u \) such that \((u, t_u) \in D\), do the following:
   - Let \( w \) be the weight of the edge \([s, u]\), set \( t_u = \min\{t_u, t_e + w\} \);
   - If \( t_u < t_{\text{max}} \) and \( D \) is not empty, then go back to (ii), else stop.

Now all that remains is to clean up \( V_P \) by removing each vertex \( u \) from \( V_P \) for which \( t_u^- > t_u^+ \). The set \( E_P \) stores all edges that connect with at least one vertex in \( V_P \).
The next step is to construct the polygons that constitute the space-time prism. This step is similar to the algorithm 3D-space-time prism described in Kuijpers and Othman (2009). As we already provided a correctness proof in that same article, we will omit to repeating it here. A major difference is that we need to correct the polygons for the anchor regions because our previous implementation neglects the internal points of the anchor region, illustrated in Figure 5.

To solve this issue, an extra, easy-to-compute, triangular polygon is added for each anchor region. For each starting region \( S_{bi} = (p, q, r^-, t^+, \mu, \chi) \), the polygon \( \langle (p, t^+)/C_0, (q, t^+)/C_0, \ldots \rangle \) is added as illustrated in Figure 5 on the right. Likewise, for each ending region \( S_{ej} = (p, q, r^-, t^+, \mu, \chi) \) we add the polygon \( \langle (p, t^+), (q, t^+), \ldots \rangle \).

### 3D SPACE TIME PRISM

**Input:** \( (\mathbb{R}^N', \{(t_u, t_u^+) | u \in \mathbb{R}^N\}', V_P, E_P) \)

**Output:** drawing of \( \mathbb{P}^\mathbb{R}N(t_0, p, t_q, q, \ldots) \).

For each edge \((r, s)\), with \( r = (x_r, y_r) \) and \( s = (x_s, y_s) \), in \( E_P \), we do the following for \( r \) and \( s \). There are three possible cases one needs to consider and they are illustrated in Figure 6.

- **Cases 1 and 2** are illustrated in Figure 6 on the left and in the middle: If \( w_{rs} \leq (t_r^+ - t_r^-)/2 \), then draw the polygon \( \langle (t_0, x_0, y_0), (t_r^+, x_r, y_r), (t_r^-, x_r, y_r), (t_0, x_0, y_0) \rangle \), where \( t_0 = (t_r^- + t_r^+)/2 \) and

![Uncovered region](image1.png)

**Figure 5.** The omitted part of an anchor region.

![Cases 1, 2 and 3](image2.png)

**Figure 6.** Cases 1, 2 and 3.
\[(x_0, y_0) = (x_r, y_r) + \frac{(t_0^+ - t_r^-)}{2} v_{rs} \frac{(x_s - x_r, y_s - y_r)}{d_{rs}}.\]

Otherwise, in Case 2, draw the polygon \(\langle (t_1, x_s, y_s), (t_r^+, x_r, y_r), (t_r^-, x_r, y_r), (t_0, x_s, y_s), (t_1, x_s, y_s) \rangle\), where \(t_0 = t_r^- + w_{rs}\) and \(t_1 = t_r^+ - w_{rs}\). These polygons capture all possible trajectories on the edge that allows travel that started in \(r\) and is able to return to \(r\) in time.

- This is Case 3, illustrated in Figure 6 on the right: In this case, an object is able to reach \(s\) from \(r\) before one has to leave \(s\) again. In other words, \(t_s^+ \geq t_r^- + w_{rs}\). In which case the polygon \(\langle (t_r^-, x_r, y_r), (t_1, x_r, y_r), (t_s^+, x_s, y_s), (t_0, x_s, y_s) \rangle\) is drawn where \(t_0 = t_r^- + w_{rs}\) and \(t_1 = t_s^+ - w_{rs}\).

- Repeat the previous steps with the indices \(r\) and \(s\) interchanged.

The result of this precomputation algorithm, together with the 3D-SPACE-TIME PRISM-algorithm is shown in Figure 7. Note that unlike in Figure 3, where a prism starts and ends in a single point, an uncertain space-time prism accounts for multiple possible locations to start from or arrive at.

Figure 7. An envelope of space-time prisms on anchor regions.
5. Measuring spatio-temporal uncertainty and flexibility with respect to anchor regions

In this section, we take things a step further and introduce the main contribution of this article. When we introduced anchor regions, we used probability functions to model the likelihood of every point in the anchor region, but we have yet to exploit these attributes of anchor regions.

Let \( r \) be any spatio-temporal point inside the envelope of the uncertain space-time prism. We know that \( r \) is covered by at least one space-time prism with a starting point in the starting region and an ending point in the ending region. Suppose, and this is the case for most points in the envelope, that there is a subset of the anchor regions consisting of points that are anchors for space-time prisms that contain \( r \). We can then measure uncertainty by integrating the probability functions over these subsets by means of the functions we introduced in Definition 4.1. Three cases are possible, we may

(i) restrict ourselves to the starting regions,
(ii) restrict ourselves to the ending regions,
(iii) multiply the two numbers above.

In all three cases, a number between 0 and 1 is obtained.

In the first case, we obtain the likelihood that the anchor of a space-time prism that contains \( r \) is part of the starting regions. In the second case, we obtain the likelihood that the anchor of a space-time prism that contains \( r \) is part of the ending regions. In the third case, we obtain the simultaneous likelihood that the anchor of a space-time prism that contains \( r \) is part of the starting and ending regions. The fraction of the anchor regions having a starting and ending point of a space-time prism that contains that particular space-time point \( r \) yields a measure of uncertainty to reach that point.

**Definition 5.1:** Let \( r \) be a spatio-temporal point on a road network \( RN \) and \( S_b, S_e \) anchor regions on \( RN \).

- The **emanating fraction of \( r \) with respect to \( S_b \)** equals the measure, with respect to the distribution functions of \( S_b \), of the subsets of \( S_b \) that contain anchors, with a smaller time coordinate than \( r \) of space-time prisms on \( RN \) that contain \( r \).
- The **absorbing fraction of \( r \) with respect to \( S_e \)** equals the measure, with respect to the distribution functions of \( S_e \), of the subsets of \( S_e \) that contain anchors, with a larger time coordinate than \( r \) of space-time prisms on \( RN \) that contain \( r \).
- The **fraction of \( r \) with respect to travel from \( S_b \) to \( S_e \)** equals the product of (the emanating fraction of \( r \) with respect to \( S_b \)) with (the absorbing fraction of \( r \) with respect to \( S_e \)).

In the following section, we show the surprisingly simple, i.e., polygonal, shape of these subsets. We also provide an algorithm to compute these subsets and associated fractions as defined in Definition 5.1.

5.1. Algorithm

The algorithm is based on the following observations, which do not require proof.
Because of the additive nature of integrals, i.e., an integral over a surface equals the sum of integrals over disjoint subsets that cover the surface, we can treat each of the rectangular regions in space-time separately and add them together once the computation is done. An anchor region can thus be seen as the sum of anchor regions on straight edges of the road network.

Let \( r = (t_r, x_r, y_r) \) be the space-time point for which we wish to compute the fraction of space-time prisms that contain that point. For each of those separate rectangular anchor regions we can distinguish between two cases. Either there exists a point \( s \) in the spatial part of the anchor region for which there exists more than one fastest path to \( r \), or there does not. If this point exists, we simply divide the rectangular region into two new regions along the temporal line through \( s \). This operation ensures we are again in the latter case, where there exists no spatial point, in the interior of the region, for which there exists more than one fastest path to \( r \) as illustrated in Figure 8.

The second observation, which will be proven in Theorem 5.1, states that it suffices to find all space-time paths that originate in the starting anchor region and end in \( r \), and all space-time paths that originate in \( r \) and arrive in the ending anchor region.

**Theorem 5.2:** There exists a space-time prism \( P_{\text{RN}}(t_p, x_p, y_p, t_q, x_q, y_q) \) that contains \( r \) if and only if there exists a trajectory from \((t_p, x_p, y_p)\) to \((t_q, x_q, y_q)\) on \( \text{RN} \) through \( r \).

**Proof:** The proof is trivial, the equivalence holds by the very definition of a space-time prism.

Although the proof is trivial, it holds the key to our algorithm. Assume that the interior of the anchor region \( S_b \) does not contain a point with more than one path with the same road network time to \( r \). We will show in the algorithm below how to split the anchor regions to smaller regions that satisfy that condition:

(i) First, a fastest path that leads to \( r \) and contains the edge contained by \( S_b \) either intersects \( S_b \), goes over \( S_b \), i.e., all its time coordinates are larger than those of \( S_b \), or is under \( S_b \), i.e., all its time coordinates are smaller than those of \( S_b \). If it goes over \( S_b \), i.e., all its time coordinates are greater than those of \( S_b \), then all points of \( S_b \) clearly have a path to \( r \), because any moving object in a point in \( S_b \) can just wait until its time coordinate equals that of the fastest path and leave to \( r \) following that path. Likewise, if the path goes under \( S_b \), i.e., all its time coordinates are smaller than those of \( S_b \), then no moving object departing from \( S_b \) is able to reach \( r \) in time. If the path intersects \( S_b \), then all space-time points of \( S_b \) with a time coordinate smaller

![Figure 8. A divided anchor region.](image-url)
than the point on the path with the same spatial coordinates have a path that reaches \( r \) in time. This is depicted in the shaded region in Figure 9a.

(ii) Second, a fastest path that emanates from \( r \) and contains the edge contained by \( S_e \) either intersects \( S_e \), goes over \( S_e \) or is under \( S_e \). If it goes under \( S_e \), i.e., all its time coordinates are smaller than those of \( S_e \), then all points of \( S_e \) can be reached from \( r \), because any moving object that departs from \( r \) can reach a point with spatial coordinates in \( S_e \) and wait until its time coordinate equals that of a point in \( S_e \). Likewise, if the path goes over \( S_e \), i.e., all its time coordinates are greater than those of \( S_e \), then no moving object departing from \( r \) is able to reach \( S_e \) in time. If the path intersects \( S_e \), then all space-time points of \( S_e \) with a time coordinate greater than the point on the fastest path with the same spatial coordinates have a path from \( r \) that can be reached in time. This is depicted in the shaded region in Figure 9b.

These observations are necessary and sufficient to compose the algorithm. Once these areas have been determined, we can integrate the distribution functions over these areas and compute a probability for a specific space-time point. The polygonal nature of these areas allows easy computation of these integrals.

Let \( r = (t_r, x_r, y_r) \) be the space-time point for which we wish to compute the fraction of space-time prisms covering it. Let \( S_b \) be the starting anchor region and \( S_e \) be the ending anchor region. Let \( S_{b,i} \) be the restriction of \( S_b \) to a single edge and \( i \) be a natural number to count the number anchor regions on \( R \) that \( S_b \) contains; the definition of \( S_{e,i} \) is analogous. The first step in the algorithm is to compute appropriately sized anchor regions. These are regions where each point in the spatial interior has a unique fastest path to the given space-time point \( r \).

The following algorithm takes a spatio-temporal point \( r = (t_r, x_r, y_r) \) as input and outputs three numbers: the emanating fraction \( S_{b,r} \) of \( r \) with respect to \( S_b \), the absorbing fraction \( S_{e,r} \) of \( r \) with respect to \( S_e \) and the fraction \( S_r \) of \( r \) with respect to travel from \( S_b \) to \( S_e \).

**Initialization.** Set \( S_{b,r} \) and \( S_{e,r} \) equal to 0.

**Step 1:** Let \( S_{b,i} = (p, q, t_{pq}, t_{pq}^+, v_{pq}, z_{pq}) \) where \( p = (x_p, y_p) \), \( q = (x_q, y_q) \). The following needs to be repeated for each \( S_{b,i} \). Let \( d_p = t_r - d_{RN}((x_r, y_r), (x_p, y_p)) \), \( d_q = t_r - d_{RN}((x_r, y_r), (x_q, y_q)) \), \( d_{pq} = d_{RN}((x_p, y_p), (x_q, y_q)) \) and \( v_{pq} \) be the reigning speed limit on the segment that supports \( S_{b,i} \).

![Figure 9. Subsets of anchor regions that are possible to connect to a fixed space-time point.](image-url)
• **Case 1:** \( d_p + d_{pq} = d_q \)

  - If \( d_p - d_{pq} = d_q \), then interchange the roles of \( p \) and \( q \), now we have \( d_p + d_{pq} = d_q \).
  - If \( d_q \leq t_{pq}^- \), then do nothing and move on to the next \( S_{b,i} \). See Figure 10 on the left.
  - If \( d_p \geq t_{pq}^+ \), then let replace \( S_{b,r} \) by \( S_{b,r} + \int_0^1 \mu p(z) \) and move on to the next \( S_{b,i} \). See Figure 10 on the right.

• **Else If** \( d_p \leq t_{pq}^- \), then let \( A \) be the area bounded by the polygon \( \langle (d_p, x_p, y_p), (d_p, x_q, y_q), (d_q, x_q, y_q) \rangle \) else let \( A \) be the area bounded by the polygon \( \langle (d_p, x_p, y_p), (t_{pq}^-, x_p, y_p), (t_{pq}^-, x_q, y_q), (d_q, x_q, y_q) \rangle \). Replace \( S_{b,r} \) by \( S_{b,r} + \int_A \mu p(z) \). See Figure 11 on the left for the first polygon and on the right for the second polygon.

Note that although \( A \) is likely to exceed the boundary of \( S_{b,i} \), the integral is still well defined because \( \mu p \) is zero outside \( S_{b,i} \).

• **Case 2:** \( d_p \pm d_{pq} \neq d_q \)

  If this is the case we have to compute the point on the segment where the two fastest paths from this segment to \( r \) intersect and distinguish three separate sub-cases. The first sub-case is the easiest, when \( d_p, d_q \leq t_{pq}^- \) then this inequality will also hold for the time-coordinate where the two fastest paths from \( r \) intersect, in that case \( r \) is not reachable from any point of \( S_{b,i} \) in time.

  - If \( d_p, d_q \leq t_{pq}^- \), then do nothing and proceed to the next anchor region. This is depicted in Figure 12 on the left.
  - If all time coordinates satisfy \( (d_p + d_q - d_{pq})/2, d_p, d_q \geq t_{pq}^- \), \( r \) is reachable from any point of \( S_{b,i} \) in time. In that case it is pointless to compute intersections and divide the anchor region. Hence, if \( (d_p + d_q - d_{pq})/2, d_p, d_q \geq t_{pq}^- \), then replace \( S_r \) by \( S_r + \int_0^1 \mu p(z) \) and proceed to the next anchor region. This is depicted in Figure 12 on the right.

In the remaining case, at least one of the fastest paths from \( r \) intersects \( S_{b,i} \) and we need to split \( S_{b,i} \) into two regions such that all points in those regions have a unique
fastest path to $r$. We apply the same strategy as in Kuijpers and Othman (2009). As shown in Figure 13, we merely need to reduce our computations to the two-dimensional case, compute the $x$-coordinate of the intersection and multiply that by an appropriate unit vector. In this case $S_{b,i}$ will be split at the spatial point

$$s = (x_s, y_s) = (x_p, y_p) + \left( \frac{d_p - d_q + d_{pq}}{2/v_{pq}} \right) \cdot \frac{(x_q - x_p, y_q - y_p)}{\sqrt{(x_q - x_p)^2 + (y_q - y_p)^2}}$$

- **Else** Replace $S_{b,i}$ by the two new regions $S'_{b,i}$ and $S''_{b,i}$ where

$$- S'_{b,i} = (p, s, t_{pq}^-, t_{pq}^+, \mu_{pq}, \nu_{pq}, \sigma) \quad \text{where}$$
Figure 13. Splitting an anchor region.

\[ f(\lambda) = \lambda \cdot \sqrt{\frac{(x_s - x_p)^2 + (y_s - y_p)^2}{(x_q - x_p)^2 + (y_q - y_p)^2}} \]

\[ - S''_{b,l} = (s, q, t_{pq}^+, t_{pq}^-, \mu_{pq}, \lambda_{pq} \circ g) \] where

\[ g(\lambda) = \lambda \cdot \sqrt{\frac{(x_s - x_p)^2 + (y_s - y_p)^2}{(x_q - x_p)^2 + (y_q - y_p)^2}} + \sqrt{\frac{(x_s - x_p)^2 + (y_s - y_p)^2}{(x_q - x_p)^2 + (y_q - y_p)^2}} \]

and store \( d_p \) in the vertex \((x_p, y_p)\), \( d_q \) in the vertex \((x_q, y_q)\) and \((d_p + d_q - d_{pq})/2\) in the vertex \((x_s, y_s)\) to avoid re-computation in Case 1. Now proceed as in Case 1 for each of these two new regions.

Step 2: The procedure \( S_{e,i} \) is very much like the one outlined in Step 1. However, as indicated in Figure 9, we need to construct the polygons on the upper side of the fastest path.

Let \( S_{e,i} = (p, q, t_{pq}^+, t_{pq}^-, \mu_{pq}, \lambda_{pq}) \) where \( p = (x_p, y_p) \), \( q = (x_q, y_q) \). The following needs to be repeated for each \( S_{e,i} \). Let \( d_p = t_r + \Delta_{\text{RN}}((x_r, y_r), (x_p, y_p)) \), \( d_q = t_r + \Delta_{\text{RN}}((x_r, y_r), (x_q, y_q)) \), \( d_{pq} = \Delta_{\text{RN}}((x_p, y_p), (x_q, y_q)) \) and \( r_{pq} \) be the reigning speed limit on the segment that supports \( S_{e,i} \).

- **Case 1:** \( d_p \pm d_{pq} = d_q \)
  - If \( d_p - d_{pq} = d_q \), then interchange the roles of \( p \) and \( q \), now we have \( d_p + d_{pq} = d_q \).
  - If \( d_p \geq t_{pq}^+ \), then do nothing and move on to the next \( S_{e,i} \).
  - If \( d_q \leq t_{pq}^- \), then let \( S_{e,r} \) be the area bounded by the polygon \( \langle (d_p, x_p, y_p), (d_q, x_q, y_q) \rangle \) else let \( A \) be the area bounded by the polygon \( \langle (d_p, x_p, y_p), (t_{pq}^+, x_p, y_p), (t_{pq}^-, x_q, y_q), (d_q, x_q, y_q) \rangle \). Replace \( S_{e,r} \) by \( S_{e,r} + \pi \mu_p \).

Note that although \( A \) is likely to exceed the boundary of \( S_{e,i} \), the integral is still well defined because either \( \mu_p \) is zero outside \( S_{e,i} \).
Case 2: \( d_p \pm d_{pq} \neq d_q \)

If this is the case we have to compute the point on the segment where the two fastest paths from this segment to \( r \) intersect and distinguish three separate sub-cases. The first is the easiest, when \( d_p, d_q \leq t_{pq}^- \) then this inequality will also hold for the time-coordinate where the two fastest paths from \( r \) intersect, in that case no moving point leaving from \( r \) will be able to reach any point of \( S_{e,i} \) in time.

If \( d_p, d_q \geq t_{pq}^+ \) then do nothing and proceed to the next anchor region.

If all time coordinates satisfy \( (d_p + d_q + d_{pq})/2, d_p, d_q \leq t_{pq}^+ \) then any moving point starting from \( r \) will be able to reach all points in \( S_{e,i} \) in time. In that case it is pointless to compute intersections and divide the anchor region. Hence,

If \( (d_p + d_q + d_{pq})/2, d_p, d_q \leq t_{pq}^+ \) then replace \( S_{e,r} \) by \( S_{e,r} + \int_0^{t_{pq}^+} \gamma_{pq} \) and proceed to the next anchor region.

In the remaining case one of the fastest paths from \( r \) intersects \( S_{e,i} \) and we need to split \( S_{e,i} \) into two regions such that all points in those regions have a unique fastest path to \( r \). We apply the same strategy as in Kuijpers and Othman (2009). As illustrated in Figure 13, we merely need to reduce our computations to the two-dimensional case, compute the \( x \)-coordinate of the intersection and multiply that by an appropriate unit vector. In this case \( S_{e,i} \) will be split at the spatial point

\[
s = (x_s, y_s) = (x_p, y_p) + \left( \frac{d_q - d_p + d_{pq}}{2/v_{pq}} \right) \cdot \frac{(x_q - x_p, y_q - y_p)}{\sqrt{(x_q - x_p)^2 + (y_q - y_p)^2}}
\]

Else Replace \( S_{e,i} \) by the two new regions \( S'_{e,i} \) and \( S''_{e,i} \) where

\(- S'_{e,i} = (p, s, t_{pq}^+, t_{pq}^+, \mu_{pq}, \gamma_{pq} \circ f) \) where

\[
f(\lambda) = \lambda \cdot \sqrt{(x_s - x_p)^2 + (y_s - y_p)^2} \quad \frac{(x_q - x_p, y_q - y_p)}{(x_q - x_p)^2 + (y_q - y_p)^2}
\]

\(- S''_{e,i} = (s, q, t_{pq}^+, t_{pq}^+, \mu_{pq}, \gamma_{pq} \circ g) \) where

\[
g(\lambda) = \lambda \cdot \sqrt{(x_q - x_s)^2 + (y_q - y_s)^2} \quad \frac{(x_q - x_p, y_q - y_p)}{(x_q - x_p)^2 + (y_q - y_p)^2}
\]

and store \( d_p \) in the vertex \((x_p, y_p)\), \( d_q \) in the vertex \((x_q, y_q)\) and \((d_p + d_q - d_{pq})/2 \) in the vertex \((x_s, y_s)\) to avoid re-computation in Case 1. Now proceed as in Case 1 for each of these two new regions.

Output \( S_{b,r} \), \( S_{e,r} \) and \( S_r = S_{b,r} \cdot S_{e,r} \).
Now that we have developed an algorithm to calculate the uncertainty associated with reaching a single space-time point, we can construct an algorithm to visualize this for all points of an uncertain space-time prism envelope. To this end, as it is impossible to calculate and visualize uncertainty for all space-time points within the uncertain space-time prism, we divide the envelope in smaller regions, pick a representative point for each region, compute its probability and assign a suitable colour to that entire region.

The 3D-SPACE-TIME PRISM algorithm outputs a set of polygons. Again we cycle over all the edges of the road network that support such a polygon. For each such edge we intersect the polygons with a two-dimensional grid on that edge. The size of the grid depends on a pre-chosen resolution. The intersection gives again a set of polygons, where no polygon is larger than the cells of the grid. For each of those polygons we compute the centre of mass, which we will use as input for our POINTPROBABILITY algorithm. This in turn yields a number, between 0 and 1, which we associate with that centre of mass and its polygon and use it to assign an appropriate colour to the polygon.

These algorithms are implemented in Mathematica as a proof-of-concept. In Figure 14, we restricted our algorithm to compute the emanating and absorbing fraction of each spatio-temporal point on the left and right, respectively. In Figure 15, our algorithm computed the emanating fraction of each spatio-temporal point with respect to travel from the originating regions to the destination region. For the sake of simplicity, we assumed a uniform distribution on the anchor regions in all these examples.

We used shades of red to visualize the computed fraction for points inside the envelope. In the case of the emanating fraction, displayed in Figure 14 on the left, the darker the shade of red gets in a point means the more points from the starting regions can possibly reach that spatio-temporal point. Points with a lighter shade of red can be reached from fewer points of the starting regions. Note that if a distribution other than a uniform one is used, then a darker shade means it can be reached from more ‘important’ points of the starting regions; a lighter

![Figure 14. A space-time prism with the emanating fraction (left) and absorbing fraction (right) coloured in shades of red. The closer the colour comes to red, the higher the value of the fractions.](image-url)
shade means it is only reachable from few points and points with less of a ‘weight’, i.e. with a low probability.

A similar reasoning can be applied to Figure 14 on the right. Finally, Figure 15 reflects the combined information that is displayed in Figure 14, and a darker shade of red reflects a more likely position in space and time of a moving object, given the constraints of the road network and the anchor regions.

6. Notes on complexity

In the algorithm PRECOMP, the bottleneck is the single source-shortest path algorithm. Because the road network is a sparse graph in practice, this algorithm can have complexity $O(n \log n)$ where $n$ equals the number of vertices in the road network. Let $s$ equal the total number of anchor regions, then the complexity of PRECOMP is $O(s \cdot n \cdot \log n)$ in the worst case.

The complexity of the 3D-SPACE-TIME PRISM-algorithm equals the total number of edges in the worst case, because the graph is sparse and represents a typical road network, the worst-case complexity equals $O(n)$.

The complexity of POINTPROBABILITY is $O(s)$. The distance computations in POINTPROBABILITY can be done in constant time as they depend on values that are computed in PRECOMP, provided we store those values. The total complexity depends on the pre-chosen resolution and is beyond the scope of this article. It would require an estimate of the surface of the envelope of the prism, which is material for future work.
7. Analysing prism uncertainty and flexibility using anchor regions

In the introduction to this article, we noted that space-time prisms and their intersections are being used in a wide range of application domains, including mobile objects databases and related location-based services, transportation planning, epidemiology, social research and crime analysis. In all of these domains, we are faced with the problem of wanting to know where an object or person could be during a designated time window given observed or known mandatory locations and times framing that time window. In some applications these observed or known anchors can have measurement error, such as the case with LATs or self-reported data. In these applications, we are interested in assigning confidence levels to the resulting prism or prism–prism intersection. In other applications we may be interested in the relationship between anchor flexibility and potential movement, activity and interaction. For example, there are policy questions surrounding the effects of increasing available hours for public services, liberalizing retail trading hours or ‘flex-time’ work schedules on improving accessibility to other opportunities. This section of the article discusses the use of anchor regions in analysing prism uncertainty and flexibility.

7.1. Measurement errors and space-time prisms

A common strategy in error analysis is to model errors on measured values and analyse how these propagate onto derived values. In time geography, we are concerned with the propagation of errors from the prism anchors to the prism itself. The methods in this article support two types of prism uncertainty analysis.

In Section 4.3, we presented an algorithm for computing an uncertain space-time prism: this is the envelope of the union of all space-time prisms that connect a point in the starting anchor regions to a point in the ending anchor regions. If the anchor regions reflect uncertainty about the prism anchors’ locations and/or times, then this envelope bounds all network locations in space and time where the person can be given this uncertainty. Querying to capture all possibilities for an uncertain mobile object is core in mobile objects databases (Sistla et al. 1998, Moreira et al. 1999); the uncertain prism extends this semantic to the space-time prism.

It is also possible to refine this approach if one has confidence regions for the anchors’ locations or times. Recall that an anchor region is bounded subset of space-time of all possible locations of an anchor point (prism anchor). This is essentially a 100% confidence region. Given a probability distribution for an anchor’s location and timing within the anchor region (including a uniform distribution), under a stated level of confidence the resulting region will also be box-shaped under the space-time independence assumption. Therefore, one can treat the confidence region as an anchor region and compute the uncertain prism using the algorithm in Section 4.3. The resulting envelope is the uncertain prism at the stated level of confidence. Note that we cannot use a simultaneous confidence region for both space and time because those regions are usually elliptical in nature. Again, relaxing the space-time independence assumption is a topic for further investigation.

7.2. Measuring flexibility

Uncertainty and flexibility are two sides of the same coin: the anchor region for an anchor point can also be interpreted as a measure of the anchor’s flexibility in space and time. The anchor region can show the locations and times where the anchor can be and still result in a prism that contains a given space-time location. In other words, given a location in space-time, where and
when can a prism’s anchors exist and will the prism still contain that location? This is a measure of the prism’s location and timing flexibility given other parameters such as the maximum travel velocity and time budget. A highly constrained prism (e.g., one with a short-time budget and low travel velocity) would have little or no flexibility with respect to its anchor locations and times, whereas a less-constrained prism (e.g., longer time budget and higher travel velocities) may have a large number of locations and times where its anchors could exist and still contain a given location. As far as we know, there is no such measure of prism flexibility in the time geographic literature, although authors have commented on the need to treat prism anchors as flexible rather than fixed events for some types of travel and activity (Schwanen 2006).

At the end of the previous section we generated a fully coloured prism where all spatio-temporal points indicated their fraction simultaneously. Assume that in a point \( r \) the emanating fraction of \( r \) with respect to \( S_b \), as defined in Definition 5.1, is close to 1. This means that we can reach \( r \) from most of \( S_b \). More importantly, this means there exists a path from \( S_b \) that reaches the spatial component of \( r \) and allows us to spend a time at this location that equals almost all of the temporal width of \( S_b \). We illustrate this in Figure 16. The shaded part of \( S_b \) covers almost all of \( S_b \), which means the emanating fraction of \( r \) with respect to \( S_b \) is close to 1. A fastest path from any point on the bottom of \( S_b \) to the spatial location of \( r \) would be parallel in space-time to the fastest path drawn in Figure 16. Moreover, suppose the temporal height of \( S_b \) equals \( \Delta t \), then any such path to the right of \( s \) from the bottom of \( S_b \) arrives at least \( \Delta t \) earlier at the spatial location of \( r \), we can thus spend at least \( \Delta t \)-time at the spatial location of \( r \). If we leave from the left of \( s \), we have a little less than \( \Delta t \)-time to spend. So, the higher the emanating fraction, the more flexibility we have to choose when and where to leave from \( S_b \) and vice versa.

Likewise, assume that in a point \( r \) the absorbing fraction of \( r \) with respect to \( S_c \), as defined in Definition 5.1, is close to 1. This means that we can reach most of \( S_c \) from \( r \). More importantly, we can spend an amount of time, which equals almost all of the temporal width of \( S_c \), at the spatial component \( r \) before we have to leave again and still be able to reach \( S_c \) in time.

This gives us a measure of flexibility we have with respect to the starting and ending regions at each location at each moment in time. A more practical way is to apply a kind of spatial projection on the road network. If we project, for each location, the maximum of these fractions in time onto the road network, we immediately obtain the flexibility we have to reach those locations with respect to our schedule or probable locations to leave from or arrive at.

![Fastest path to r](image)

Figure 16. Illustration of flexibility.
8. Conclusions and future work

This article relaxes the strict theoretical assumption of rigid and perfectly known anchor points in network-based space-time prisms. Classical space-time prisms are defined on the basis of two fixed anchor points showing locations and times where a person or object must be at the beginning and end of the prism, and strictly classify locations as either accessible or inaccessible. This is unrealistic in many real-world applications where prism anchors have measurement error or inherent flexibility. To make time geography’s key concept of the space-time prism more realistic, we have introduced the concept of an uncertain space-time prism based on anchor regions. This concept extends and generalizes the classical space-time prism by representing the prism anchors as a set of possibly disconnected locations and times with associated anchor probabilities. We develop two algorithms for calculating network-based space-time prisms based on probabilistic anchor regions. The first algorithm calculates the envelope of all space-time prisms having an anchor point within a particular anchor region. The second algorithm calculates, for any space-time point, the probability that a space-time prism with given anchor regions contains that particular point. Both algorithms have reasonable worst-case complexity.

The contribution of this article is at least threefold. First, our study enhances the current practice of space-time accessibility analysis. Empirical studies of individual space-time accessibility typically derive the anchor points of space-time prisms from the start and end points of fixed activities reported in activity-travel diaries (Weber and Kwan 2002, Kim and Kwan 2003b, Neutens et al. 2008b). Although there is much to say for this pragmatic approach, a drawback is that the revealed activity timing of fixed activities at a particular day is indiscriminately considered as a hard constraint: the approach does not allow the sampled individual to specify flexible boundaries during which an activity should start or end. However, what if an individual has staggered working hours or may choose to work at home at a given day? These considerations may significantly affect the number of urban opportunities accessible to an individual and thus the assessment of space-time accessibility. Given the increased number of telecommuters and individuals having flexible working schedules, this has become an increasingly harmful limitation to the classic space-time prism. Current accessibility measures based on space-time prisms cannot deal with such issues. In this respect, the uncertain space-time prism presented in this article provides an essential first step towards the development of individual accessibility measures that account for spatial and temporal flexibility of activity engagements.

Second, we have enhanced the representational realism of space-time prisms. Although classical three-dimensional prism representations have rather illustrative value, the uncertain space-time prism generated by our procedure provides a comprehensive synopsis of an individual’s travel possibilities if start and end locations and times have uncertainty. To further increase the realism of the uncertain space-time prism, we have defined the concept within a road network. The result is a three-dimensional representation comprising a set of polygons depicting where and for how long an individual can be on the road network. The colours of the polygons represent how likely it is that the individual is able to reach a particular space-time point.

Third, the analytical foundation of the uncertain prism also contributes to research seeking to add rigour to the classically conceptual framework of time geography (Miller 2005b). The dichotomy between fixed and flexible activities is central to time geography: the former being the prism anchors and the latter the potential opportunities being modelled by the prism. The strict dichotomy between these activity types, although very useful, is artificial: it is often the case that a given activity can have degrees of fixity versus flexibility. The theory and mathematical procedures presented in this article extend the analytical foundation of time geography to encompass a more general definition of anchor points that can include arbitrary degrees of flexibility.
The approach presented in this article opens up the use of time geography to a wide range of practical applications. Our approach may, for example, be useful for transportation studies. As the uncertain space-time prism indicates where and for how long individuals can be on a road network, the combination of two such prisms is an insightful instrument to identify potential carpool partners and suggest potential meeting locations for ride-sharing. Our approach is particularly suited for revealing the travel possibilities while accounting for variability in working hours and workplaces. More generally, the probability measures elaborated in this article may support travellers in making decision where and for how long they may pursue a particular activity while coping with unreliable travel and potential changes to their activity schedule. The uncertain space-time prism has also practical relevance for crime research. The concept might, for example, be used to check a suspect’s alibi based on uncertain statements of witnesses about the time and place where the suspect has been spotted. Another example would be to set roadblocks for fugitives who escaped from a loosely defined region during a certain time span.

Some additional research and development is required before uncertain time prisms can be applied in a wide range of domains. An immediate task is implementation of the basic approach in spatial databases and GIS software. This current article focuses on the development, demonstration and dissemination of the theory and methodology to the GIScience community. Implementation using spatial database and GIS software, although not trivial, is not in question: this article provides the detailed mathematical procedures, including correctness proofs, and demonstrates the tractability of required calculations. Implementation issues concern questions such as the appropriate data structures, database indices and other detailed algorithmic design issues, heuristics (if any) needed for scalability to large applications, as well as the need for effective query languages and interface design to support user interaction with the tools and their results.

An important extension of our approach would be to deal with uncertainty in other prism parameters, such as the travel velocity and the stationary time. Also, the question as to how to determine the probability functions of the anchor region has remained unexplored in this article and remains an interesting challenge for future work.

The complexity analysis we provided is a worst case one. In the worst case, at least one temporal interval is large enough so that all nodes in the road network can be reached, in this case the run-time of the precomputation algorithm is $O(n \log n)$. However, the extra stop-condition we inserted ensures that we do not always reach that upper bound. Consider the case where the maximal temporal separation between sampled regions is at most $\tau$, and that a moving object cannot reach the entire road network in time $\tau$ but only a fraction of $m$ nodes of the network. This would rely on defining suitable heuristics that relate the width of temporal intervals $\tau$ to the number of possible distinct edges that can be travelled, which in turn also relies on network density. But these are all aspects we left out in our simplified model and can be tackled in future work.

In the simplest case, when all distributions on space and time are uniform, the uncertain prism has a lot of symmetry. Further study is needed to exploit this symmetry and speed up the computation. In this article we restricted our anchor regions to box shapes; this can easily be extended to other shapes because the intersection we need to compute remains the same, i.e., the intersection of one or two half-spaces with the anchor region.

Another aspect that has not been studied in this article is a measure to rank a space-time point’s probability inside a single prism. Intuitively, one can imagine that points near the edge of the prism are less likely that points on the interior, however, a suitable likelihood function to express this intuition has not been found. A logical step is then to figure out a way to combine both measures and interpret them in a sensible manner.
Acknowledgements
This research has been partially funded by the European Union under the FP6-IST-FET programme, Project no. FP6-14915, GeoPKDD: Geographic Privacy-Aware Knowledge Discovery and Delivery, and by the Research Foundation Flanders (FWO-Vlaanderen), Research Project G.0344.05.

Notes
1. These edge embeddings may intersect in non-vertex points. So, we can model bridges and tunnels in our model.
2. We mean the single-pair shortest-path distance that is commonly used in graph theory and that can be computed efficiently by the well known Dijkstra’s algorithm (Weisstein 2007).

References


