

RESEARCH ARTICLE

# Modeling and manipulating spacetime objects in a true 4D model

Ken Arroyo Otori, Hugo Ledoux, and Jantien Stoter

3D Geoinformation, Delft University of Technology, the Netherlands

*Received: August 15, 2016; returned: September 16, 2016; revised: October 13, 2016; accepted: November 20, 2016.*

---

**Abstract:** The concept of spacetime has long been used in physics to refer to models that integrate 3D space and time as a single 4D continuum. We argue in this paper that it is also advantageous to use this concept in a practical geographic context by realizing a *true 4D model*, where time is modeled and implemented as a dimension in the same manner as the three spatial dimensions. Within this paper we focus on 4D vector objects, which can be implemented using dimension-independent data structures such as generalized maps. A 4D vector model allows us to create and manipulate models with actual 4D objects and the topological relationships connecting them, all of which have a geometric interpretation and can be constructed, modified, and queried. In this paper we discuss where such a 4D model fits with respect to other spatiotemporal modeling approaches, and we show concretely how higher-dimensional modeling can be used to represent such 4D objects and topological relationships. In addition, we explain how the 4D objects in such a system can be created and manipulated using a small set of implementable operations, which use simple 3D space and 1D time inputs for intuitiveness and which modify the underlying 4D model indirectly.

**Keywords:** 4D GIS, spacetime, spatiotemporal modeling, spatiotemporal operations, geometric modeling

---

## 1 Introduction

Space and time have long been considered to be interlinked [4]. Various mathematical models of reality are based on the concept of *spacetime*—usually a four-dimensional continuum that combines three spatial dimensions and one temporal dimension. Such integrated

spacetime models are not only used in physics to describe our reality abstractly, but as we argue in this paper, they can also be directly used to create concrete computer representations for the description of geographic phenomena.

In fact, despite very important conceptual differences between time and space [80], we believe that models in which they are *integrated* often have clear advantages for modeling time, and that is true from both theoretical and practical standpoints. Space and time are already similar as far as a computer is concerned: they are naturally parametrized into sets of coordinates, and they can both be decomposed into discrete geometric components (e.g., points, lines, and polygons in space, and moments and intervals of time) and topological components (e.g., relations of incidence, adjacency, and connectivity in space; and relations for causation and events that occur immediately before and after each other in time [6,38]). These discrete components can themselves be easily converted into a computer representation, as they are respectively translated into instances of database tuples or various data structures, and into the links connecting them. In the context of geographic information standards, this twin relation is expressed in ISO 19107 [56] and ISO 19108 [57], which respectively define analogous modeling definitions, concepts, classes, and relations for geographic phenomena in space and in time.

However, despite the clear similarities between space and time, as discussed in Section 2.1 most spatiotemporal modeling approaches opt to model space and time differently and separately. In most cases, this separate strategy involves relatively independent space-centric and/or time-centric models that capture specific regions at various times, a heavy use of timestamps, a description of various events, and links between corresponding or related objects across separate models. These models make it possible to represent space and time, but there is often information that is difficult to store, such as the state of a region between two separate timestamped models, the equivalences between the related objects at different times, or the topological changes that occur in these objects. In addition, applying complex queries in these models is often cumbersome or highly inefficient, such as for connectivity and topological queries across time.

Rather than following this type of separate approach to spatiotemporal modeling, we believe that for certain applications there are significant advantages in integrating 3D space and time into a single true 4D model populated by up-to-4D objects. This means that 3D objects existing along a span of time are not only conceptually 4D, but are modeled as actual 4D objects with 4D geometries and 4D topological relationships (that is, topological relationships between geometric primitives of dimensions of up to four). While lower-dimensional objects embedded in 4D (e.g., trajectories and point clouds) can be also stored and analyzed natively using this method, it is a rather inefficient way to do so. This manuscript thus focuses on the *higher-dimensional modeling of 4D spatiotemporal vector objects*.

By following this approach, it is possible to store the state of every object of every dimension at all times, as well as to handle and store the geometry and topology of the model in an explicit, generic, and unified manner [10]. Even complex situations where objects move and change shape simultaneously can be handled correctly, equivalences between objects are always explicitly stored, and attributes can be attached to all geometries of any dimension and all the topological relationships between them. This makes it possible to query, for instance: whether two objects were ever connected; which was the best possible path in a model at any given previous point in time; or to perform checks on a model using simple 4D constraints, such as safety checks that ensure that a structural element in a construction model is put in place after those that it depends on.



As background for the rest of the paper, we first present in Section 2.2 the concept of time as a dimension and how it has been reflected into some spatiotemporal models and applications. Section 2.3 then contains a summary of  $n$ -dimensional cell complexes, which provide the concepts and the terminology that is used afterwards when referring to 4D spatiotemporal objects. Afterwards, Section 2.4 describes some of the spatiotemporal operations that have been proposed by different authors and a few of their notable applications.

Then, we present the two core topics of this paper: (i) how higher-dimensional modeling can be used to model 4D space and time as a true 4D model consisting of up-to-4D vector objects in Section 3, and (ii) how these objects can be manipulated in practice using a small set of operations in Section 4.

Higher-dimensional models and the operations on them can certainly be complex, but by presenting these two aspects, we hope to show that they can nevertheless remain intuitive and implementable. Moreover, we show how users can easily model and manipulate 4D spacetime objects *indirectly* by performing relatively simple operations on the different dimensions independently, such as by manipulating sets of 3D objects in standard 3D modeling software or temporal intervals directly passed as parameters. This approach makes it possible to keep the expressive power of a 4D model but it obviates the need for a user to manipulate the primitives in the 4D model directly—a complex and highly error-prone operation. Finally, we finish with some conclusions and a short discussion in Section 5.

## 2 Related work

### 2.1 Spatiotemporal models with distinct space and time modeling

There are a great number of different spatiotemporal models that have been developed over the years. However, upon closer inspection most of them are variations on a few general approaches, and many of them are based on extensions of the existing spatial models available in commercial GIS.

We therefore summarize here the main approaches we have found together with their earliest known usage (to us) and some other prominent examples. For a thorough review of spatiotemporal models see Al-Taha et al. [5], Peuquet [81], or Pelekis et al. [78].

It should be noticed that, as mentioned in the Introduction, we focus in this paper on the modeling, storage, and analysis of objects that have a volume in 3D. When these objects exist along an interval of time and are embedded in 4D, they generally become four-dimensional objects. We do not restrict ourselves to space partitioning, and cases such as that on Figure 1(b) are allowed. This implies that models to manage lower-dimensional objects, such as trajectories and point clouds, are not further discussed in this section. Trajectories are, in most cases, lines that are embedded in 4D, and each point in a temporal point clouds is similarly a point embedded in 4D. While these can in theory be stored and analyzed using the method we present in this paper, it is not efficient to do so. For trajectories, the reader is referred to tailored solutions, e.g., Gudmundsson et al. [47] and Horne et al. [53]. We also ignore explicitly dynamic field representations stored with grids since these are better stored with  $n$ D grids [70,76,103]. However, if a field is modeled with vector-based methods, e.g., with a 3D triangulation [63] or with a Voronoi diagram [65], then the method we present in this paper can be directly applied since each 3D object become a 4D object and a subdivision of the 4D space is modeled.

**Snapshot model** The earliest mechanism to model time was rather trivial: time was simulated by showing a series of separate *snapshots*, where each snapshot is a representation of a certain region at a certain time [9]. With this model, every layer has a date associated with it and every object on this layer is considered static until the time of the next snapshot. It has many problems, often caused by the fact that objects are often represented multiple times across several layers, which causes data redundancy, possible data inconsistencies between layers (e. g., due to edits in a snapshot that are not propagated onto others), and slow queries due to the fragmentation of objects across layers. It is also impossible to know exactly when a change occurred, as it could have happened at any time between those of the two layers surrounding the event.

**Timestamps per static object** To solve these problems, other models often attach *timestamps on individual objects* rather than on entire layers, allowing for more fine-grained control without requiring separate layers for every span of time during which *all objects are static*. The objects involved might be of any dimension (e. g., polylines as in the US Historical Boundary File [18]) or polygons in a cadastral database [55]. The objects are therefore attached with a pair of timestamps, which demarcate the *start* and *end* of the period during which they existed as is represented. Objects that change in time must thus be stored as multiple representations of 2D [9] or 3D [51] structures—one for each period of time in which they did not change. Timestamps per object are usually much more efficient than those per layer in terms of storage, but they are also not very powerful: the model still does not contain explicit events, and the topological relationships between the objects vary in time and are thus hard to store explicitly. For instance, in order to answer a query in which complex topological relationships are necessary, one must often perform the query in multiple steps or reconstruct the complete state of the region at a given time. Queries in spacetime can also be very slow (e. g., finding when and where were two objects connected).

There have been related models that partially solve these problems. For instance, Gold et al. [46] and Van Oosterom [101] both allow updating the topology in an incremental manner. Representing continuous changes is thus possible with these models: local updates are performed as soon as an event occurs, and it is possible to rebuild every topological state of a map since all the operations are reversible. Another improvement is to reduce space usage by storing differential changes only [62], specifying the areas that are added to or removed from an object at a given time. Finally, a common optimization is keeping certain important snapshots explicitly (e. g., the current state of the map), which can greatly improve the response time of common queries.

**Space-time composites** In the *space-time composite* model, objects are first split into homogeneous regions that share the same history, similar to how overlapping objects are often handled in topological GIS models by computing map overlays [30,93]. Rather than the snapshot or timestamped object approaches where objects are split *temporally* into time spans with no change, space-time composites thus split objects *spatially* into regions with the exact same history (e. g., belonging to a given sequence of objects during specific time-frames). This model was first fully described in Chrisman [24] based on earlier work in Peucker and Chrisman [79]. The approach is more flexible than the snapshot model, as it allows for the explicit storage of the history of every object and the topology between the objects. However, on datasets with long and complex object histories, as well as on those





that are generated independently at each time, objects can become extremely fragmented, slowing down many operations. For instance, updating the attributes of an object might involve updating all the regions that the object is split into.

**Event-based models** Other models use events as principal entities, i. e., points in time where objects change, such as by keeping a list of changes per object [80,105]. For instance, Peuquet and Duan [82] maintain a main structure consisting of a list of events with their corresponding changes and a base map (to which these changes are applied). Unlike other models, this makes it possible to know when events exactly occurred, and to identify and attach attributes to individual changes and events (e. g., what an event represents or why it occurred). In the *history graph* model [89], different types of events are supported. Among other things, this makes it possible to model continuously changing events by specifying how data should be interpreted between events (e. g., interpolating certain values in a certain manner). A problem with event-based models is that spatial objects can often only be accessed through their related events, causing many queries to first require expensive searches in order to find objects within each event (e. g., finding the objects within a region, or finding the time spans in which a given object existed).

**Separate models for space and time** A different option is to keep track of both space and time but to do so independently, thus creating separate spatial and temporal objects and linking them appropriately. So-called *three-domain models* are based on this concept, the third domain being semantics. Examples include Yuan [110] and Claramunt and Thériault [26]. In another example, Van Oosterom [101] uses an identifier consisting of both a region identifier and a timestamp to index spatiotemporal objects.

**Conceptual models** There are also some more generic spatiotemporal models described at a more conceptual level [97], which are usually only partially specified and thus do not fit neatly in any other category. These models might not offer a complete solution, but they can be further specified and adapted to suit specific applications. For instance, Tryfona and Jensen [99] describe the space-time entity-relationship model, which is based on the entity-relationship model [23] common in database modeling. Claramunt et al. [25] discuss an object-relational model that is specifically tailored to model change, while Parent et al. [77] describe a similar one based on geometry-changing processes. Price et al. [85] describe an extension of UML where class attributes can be marked as spatial, temporal or spatiotemporal.

**Object-oriented models** In contrast to the flat hierarchies of the models above, most newer spatiotemporal models are instead *object-oriented*, using distinct spatial, temporal, and spatiotemporal classes which are then connected by various relationships, such as topological relationships, locations, events, and processes. These models are notably different from others because their entities are usually classes with their own complex internal structures, as opposed to earlier models in which they were simpler static data types. They thus incorporate object-oriented programming features such as composition, polymorphism, delegation, and inheritance. For instance, an early spatiotemporal model based on spatiotemporal objects using the IFO database model [1] is proposed in Worboys et

al. [109], another one based on a similar concept in Renolen [90], a model based on spatiotemporal simplexes as primitives in Worboys [108], and one based on distinct objects and events in Wachowicz and Healy [104].

## 2.2 Time as a dimension

Space in 2D and 3D GIS is usually considered and modeled in a manner resembling a true dimension. Some notable exceptions are 2.5D data, extruded blocks and some forms of boundary representation. However, time, as seen above, is usually treated as a simpler attribute instead. This means that while spatiotemporal objects can usually support complex geometries representing their shape, their temporal characteristics are often abstracted down to one or two values, which is geometrically equivalent to a single point or interval along the time dimension. Among other drawbacks, such a simplification limits the storage of more complex and non-static configurations, such as objects that exist along multiple periods of time, or objects that move or change their shape. It also makes it difficult to perform queries involving both space and time (such as the examples in the introduction) as all the different versions of a dataset must be reconstructed.

In order to allow for more complex (and arbitrary) configurations, it is instead possible to model time as something closer to a true dimension. By following a similar approach to how 2D/3D space is modeled, arbitrary geometries along a 1D time axis can be represented (e.g., by an arbitrary point set in  $\mathbb{R}$ ), and by extension, arbitrary combinations of 1D time geometries and 2D/3D space geometries can be represented as well. In fact, the concept of time as a dimension to be combined with the spatial dimensions is an established one with proven applications. As Couclelis [28] states, time has historically been linked to space and often considered as another dimension, and the 4D spatiotemporal manifold known as the Minkowski space [75] is used in the description of special relativity in mathematics and physics.

In the specific context of geographic information standards, ISO 19108, which covers the modeling of geographic phenomena in time, also points in this direction, stating that “time is a dimension analogous to any of the spatial dimensions” [57]. More importantly, the concepts, classes, and relationships embedded in the standard are essentially analogues of the widely implemented ISO 19107 [56], which is concerned with the description of geographic phenomena in space.

A representation where time is modeled as one more spatial dimension has been more-over used in several geographic applications, although mostly limited to 2D space and time [48]. For instance, Hornsby and Egenhofer [54] model the set of possible locations that an individual could visit as a *geospatial lifeline*—a polyline or a succession of conjoined cones in 2D space+time. Among several applications, models that consider time as a dimension have been used to analyze paths [49], to detect and describe patterns [8], to analyze motion in computer vision [21], to analyze animal movements and habitat use [36], to visualize geographic information [2, 3, 60] and to visually analyze it [7]. Alternatively, Ligozat [68] parametrizes temporal intervals into *two* dimensions, generating triangular regions of the plane which can be interpolated, analyzed, segmented or visualized [87], as well as generalized into different levels of detail [100]. 3D+time systems have also been frequently proposed [40, 44], albeit this distinction usually remains only at a conceptual level and is not reflected into most implementations. It has however been described in detail in a GIS context by Pigot and Hazelton [83], but they have to our knowledge not attempted

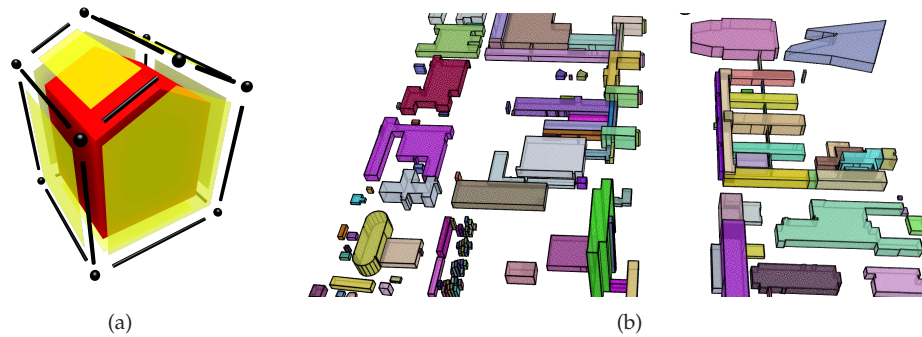


Figure 1: Two 3D cell complexes. (a) A simple model of a house contains a set of 0-cells (black balls), 1-cells (black cylinders), 2-cells (yellow polygons), and a 3-cell (red volume). The graphical representations of the 1-, 2- and 3-cells are reduced to be able to show all the cells in the complex. (b) A 3D city model with several buildings showing only its 1- and 2-cells. All the 2-cells of every 3-cell are shown in the same color.

to implement it. In GIS and other fields, a 2D space+time representation is often called a *space-time cube*.

### 2.3 $n$ -dimensional cell complexes and their computer representation

In order to describe the higher-dimensional modeling approach that we apply to 3D space and time, we use in this paper the basic concepts and notation behind *cell complexes*. Intuitively, an  $n$ -dimensional cell complex is a structure made of connected *cells* of dimensions from zero up to  $n$ , where an  $i$ -dimensional cell ( $i$ -cell),  $0 \leq i \leq n$ , is an object homeomorphic to an open  $i$ -ball (i. e., a 0D point, 1D open curve, 2D open disk, 3D open ball, etc.). Note that these objects do not contain their boundary, unlike closed  $i$ -balls which do contain it. 0-cells are commonly known as vertices, 1-cells as edges, 2-cells as faces and 3-cells as volumes. As there is no common name used for 4-cells, we simply refer to them as such. For GIS purposes and considering only linear geometries, 0-cells are used to model points, 1-cells to model line segments, 2-cells to model polygons, 3-cells to model polyhedra, 4-cells to model *polychora*, and so on. Figure 1 shows graphical representations of two cell complexes.

An  $i$ -cell ( $i > 0$ ) is bounded by a structure of  $j$ -cells,  $j < i$ , which are collectively known as its *boundary*. A  $j$ -dimensional *face* ( $j$ -face) of an  $i$ -cell is a  $j$ -cell,  $j < i$ , that is part of the boundary of the  $i$ -cell. A *facet* of an  $i$ -cell is an  $(i - 1)$ -face of the  $i$ -cell, and a *ridge* of an  $i$ -cell is an  $(i - 2)$ -face of the  $i$ -cell. Two  $i$ -cells are said to be adjacent if they have a common facet, and an  $i$ -cell and a  $j$ -cell,  $i \neq j$ , are said to be incident if either is a face of the other. A facet of a polyhedron is thus one of the polygons on its boundary, and a ridge of the polyhedron is one of the line segments on its boundary. See Hatcher [52] for a more rigorous definition of a cell complex based on induction.

Various surveys describe the data structures that can be used to represent  $n$ -dimensional cell complexes [14, 27]. Possible data structures include incidence graphs [71, 93, 96], Nef polyhedra [20], and ordered topological models [22, 66].  $n$ -dimensional generalized maps [29, 66]—a type of ordered topological model—were used for the examples shown

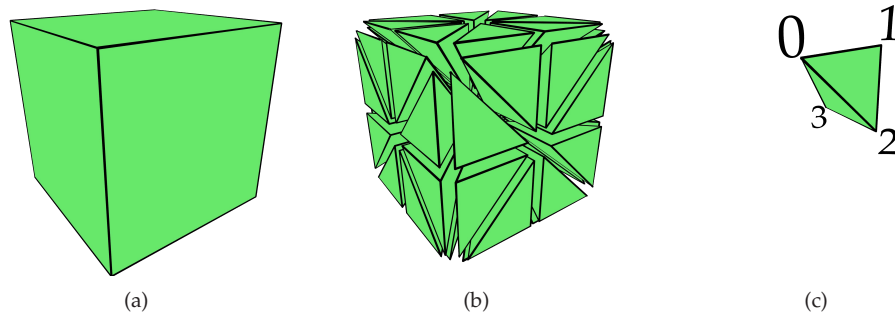


Figure 2: A possible abstract simplicial decomposition to create a generalized map. Starting from (a) a cube, (b–c) 3-simplices are formed as quadruples of an input vertex (0), edge (1), face (2) and volume (3), all of which are incident to each other.

here, as they are able to represent a wide variety of models and can elegantly handle attributes for the cells of every dimension.

In short, generalized maps model objects of any dimension by performing an abstract simplicial decomposition of the objects. Unlike in a geometric simplicial decomposition (i. e., an  $n$ -dimensional constrained triangulation [95]), the simplices do not correspond to a specific region of the object and they do not require any geometric operation in order to create them. There are various ways in which such an abstract simplicial decomposition can be performed, but for the purposes of this paper, it is similar to a barycentric triangulation of an object (Figure 2). In the 4D case, every such 4-simplex represents a unique quintuple of a vertex, an edge, a face, a volume and a 4-cell, all of which are incident to each other. See Damiand and Lienhardt [29] for more details.

## 2.4 Construction of spatiotemporal models and the operations on them

The basic operations on spatiotemporal models are usually data queries involving a combination of space and time. In fact, Langran [61] already showed that GIS queries always have time as a component, either implicitly (e. g., “now”) or explicitly. Such queries can return both spatial objects (e. g., “where does change occur?”) and temporal attributes (e. g., “when did change occur?”). However, note that for our purpose there is no such distinction—both kinds of queries would return point sets in  $\mathbb{R}^4$ .

More concretely, queries are directly based on the formal logics that define the possible relationships between spatiotemporal objects. Regarding purely *spatial relationships*, the typical GIS approach usually reduces them to a combination of set membership, adjacency, and incidence on a planar partition. In fact, the most common schemes for topology in a GIS are implemented solely on inspecting which objects are in a set [39, 48, 107]. These can be very intuitive, as they often have direct correspondences to natural language equivalents (e. g., “inside,” “touches,” “equals,” etc.) [37]. More complex relationships can also be expressed, such as those that occur between objects that are nested in each other.

Regarding purely *temporal relationships*, Allen and Ferguson [6] present the most common temporal interval logic framework used in GIS. It can be used to express relations such as: before/after, meets/met by, overlaps/overlapped by, starts/started by, dur-

ing/contains, and finishes/finished by. More complex propositions can then be assembled using propositional logic to combine multiple relations on one, some or all the objects in a dataset.

Worboys [106] describes some of these theories of logic applied to joint spatial and temporal relationships, including tense and temporal logics (propositional logics incorporating time) [86], situation calculus (finite state machines based on situations as states and actions as transitions) [72], event calculus (a form of predicate logic with events that occur, continue, are initiated and are terminated) [59], and interval temporal logic (operations on intervals and accompanying names for them) [6]. Erwig et al. [40] presents a concrete scheme of spatiotemporal queries as combinations of spatial and temporal queries, and Erwig and Schneider [41] describes a series of 2D space+time predicates analogous to predicates in 3D space. Hallot and Billen [50] present a set of 25 point-point spatiotemporal relationships that are projected to lines in a primitive space.

Peuquet [80] discusses a more direct approach, in which the spatiotemporal relationships that are stored in a computer system essentially serve as operations that are used for querying spatiotemporal datasets. She describes some of these queries, dividing them into queries about changes in spatial objects, about changes in the spatial distribution of spatial objects, and about temporal relationships between multiple geographic phenomena.

Jiang and Worboys [58] present a tree-based representation that describes a topological model for nested objects and how it changes according to various spatiotemporal events. Such changes reflect for instance the addition of a new nested object, the enveloping of an object by another, or the splitting of an object so as to remove one of its holes.

Various other spatiotemporal operations are described based on space-time cube models [49]. For instance, within a visualization context, Bach et al. [16] provides a comprehensive review of the operations that can be applied. Bach et al. [17] describes a framework where various visualization operations are described as geometric operations on a space-time cube. Miller [74] describes how accessibility-related concepts in a space-time cube can be modeled and manipulated. Demšar et al. [31] presents a method to visually analyze the density of aggregated trajectories using a volumetric representation. Demšar and Çöltekin [32] quantify the interactions between eye and mouse movements also using trajectories on a space-time cube.

Le [64] presents a method to construct spatiotemporal datasets based on interpolating 3D models at different times. It uses a combination of interactive user input (control points) and automated solving of morphing equations. Theodoridis et al. [98] describe a method to construct synthetic spatiotemporal datasets of points and modeled according to specific spatial distributions and temporal evolutions. Renolen [91] presents methods for the generalization and data reduction of spatiotemporal datasets.

### 3 Higher-dimensional modeling

The concept behind higher-dimensional modeling is to represent any number of *parameterizable characteristics* related to geographic information (e.g., two or three spatial dimensions, time, and scale) as dimensions that are modeled identically and geometrically [10]. In this manner, real-world 0D–3D entities are modeled as higher-dimensional objects embedded in higher-dimensional space, which are consequently stored using various *higher-dimensional data structures* [14].



Using this approach, an object is usually modeled in a higher dimension than its real-world counterpart when it exists along one or more *intervals* along the non-spatial dimensions. If it only exists at a finite number of *points* along these dimensions (e. g., a moment in time), the model is of the same dimension as the real-world object although it might be composed of multiple disconnected components.

The fundamentals of higher-dimensional modeling are well grounded in long-standing mathematical theories. Descartes [33] already laid the foundation for  $n$ D geometry by putting coordinates to space, allowing the numerical description of geometric primitives and the use of algebraic methods on them, theories of  $n$ D geometry were developed by Riemann [92] among others, and Poincaré [84] developed algebraic topology with a dimension-independent formulation from the start, stating that even if  $n$ D objects could not be [then] represented, they do have a precise topological definition, and consequently properties that can be studied.

The higher-dimensional modeling approach opens the door to new practical possibilities as well. From an application point of view, topological relationships between 4D objects provide insights that those between timestamped 3D objects often do not [11], such as the equivalences between multiple representations of the same 3D object and the changing topology of an object or its connectivity to others through time. Also, McKenzie et al. [73] contend that weather and groundwater phenomena cannot be adequately studied in less than four dimensions, and Van Oosterom and Stoter [102] argue that the integration of space, time, and scale into a 5D model for GIS can be used to ease data maintenance and improve consistency, as algorithms could detect if the 5D representation of an object is self-consistent and its different representations at different scales do not conflict with other objects in spacetime.

It is important to note that this higher-dimensional modeling approach bears no relation to the most common usage of 4D, 5D, 6D, ...,  $n$ D GIS/BIM in both GIS product descriptions and the scientific literature. There, such terms are generally used to refer to *any* kind of support for the storage of time information, costs, project life cycles, or any other kind of non-spatial information. In most cases, this information is simply stored as a tuple of attributes that are appended to 2D/3D objects (e. g., timestamps), or as external structures that are not linked to any objects with a geometric description (e. g., IFC scheduling information or the time series of 2D/3D objects in most software). If one were to interpret these objects geometrically, they would result in one or more 2D/3D objects embedded in higher-dimensional space, and *not in higher-dimensional objects embedded in higher-dimensional space*. It is thus that we refer to our four-dimensional modeling approach with actual 4D objects as a *true 4D model*.

However, it is worth noting that real strides towards higher-dimensional GIS have been made in various forms, including the fundamental aspects described in Section 2. In addition, research in multidimensional GIS aims at the integrated storage and analysis of heterogeneous objects of different dimensions [45,88], usually limited to 0D–3D objects but sometimes (conceptually) extended to higher dimensions.

### 3.1 Applying higher-dimensional modeling to 3D space and time

Higher-dimensional modeling can be directly applied to 3D space and time without particular concern for the specific ways in which space and time differ. Since 3D Euclidean space ( $\mathbb{R}^3$ ) is generally used to model physical 3D space and 1D Euclidean space ( $\mathbb{R}$ ) is similarly





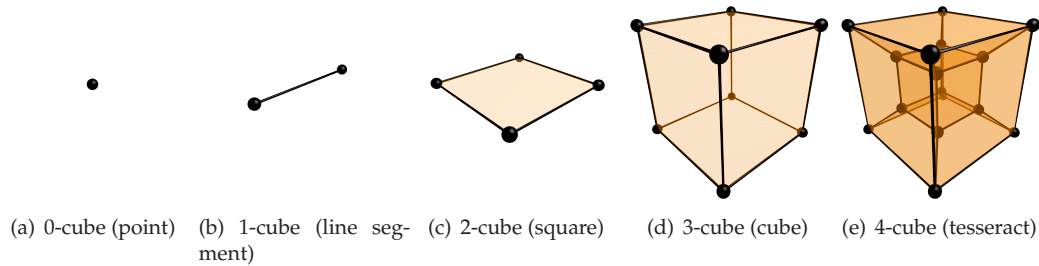


Figure 3: Cubical cells of different dimensions. An  $i$ -cube that exists along an interval of time is modeled as an  $(i + 1)$ -cube. The former can be extruded into the latter.

used to model time, it follows that 4D Euclidean space ( $\mathbb{R}^4$ ) serves as an adequate base for a 4D model integrating 3D space and time. Even non-linear models of time can be usually parameterized in a way that is mapped easily to  $\mathbb{R}$  with the help of added attributes, such as is done in implementations of various types of calendars based on the time elapsed from an epoch (e. g., Unix time).

We can thus easily define a 4D coordinate system comprised by three spatial axes  $x$ ,  $y$  and  $z$  and one temporal axis  $t$ , such that a point in 4D space can be described by a tuple of coordinates of the form  $(x_0, y_0, z_0, t_0)$ . While the order of the spatial and temporal dimensions is not important, time is generally appended at the end as the fourth dimension by convention and is thus done here as well.

### 3.2 What 3D space+time as 4D space looks like

In order to understand what a 3D space+time setting modeled as 4D space looks like, it is easier to first consider a case with a 2D space plane  $xy$  where time  $t$  is added as a third dimension. This is related to the previously mentioned *space-time cube* models and analysis as used in GIS [49]. In this configuration, a set of 2D objects, each of which exists for one or more intervals of time, is thus represented as a set of volumes. An  $i$ -cell that exists along an interval of time is thus modeled as an  $(i + 1)$ -cell. Figure 3 shows how this works using the example of the cubical cells family from dimension zero up to four.

Consider the example in Figure 4. Starting from a 2D configuration with three footprints that are each associated with a non-zero length interval of time of the form  $[t_{\text{start}}, t_{\text{end}}]$ , a 3D representation of the same footprints can be created by extruding each footprint along the time dimension.

In this 2D space+time representation, at any one moment in time (i. e., a horizontal plane slicing the 3D model perpendicularly to the 2D space plane  $xy$ ), a polygon is still represented as a polygon but one that is embedded in 3D space, parallel to the 2D space plane  $xy$  and orthogonal to the time axis  $t$ . As each of these objects remains static during a time period, they are modeled as prisms in 3D, with two identical bases that are parallel to the 2D space plane  $xy$  and all other faces being orthogonal to it. As shown in Figure 5, moving objects can be modeled by related simple shapes, such as antiprisms, twisted prisms, parallelepipeds, frustums, wedges, and pyramids, among other shapes [15].

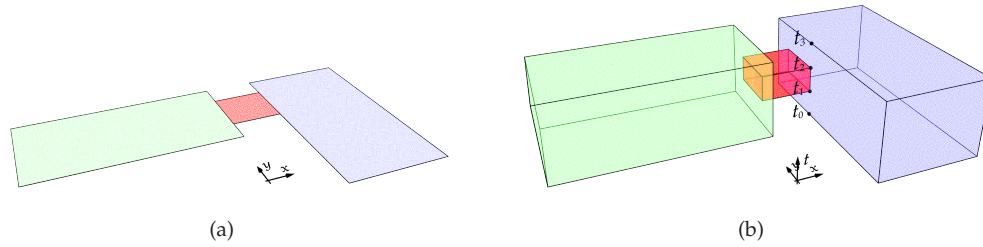


Figure 4: Starting from (a) a 2D representation of the footprints of two buildings (green and blue) and a connecting corridor (red), (b) a 3D representation is created using 2D space  $(x, y)$  + time (the vertical axis  $t$ ). The two buildings (green and blue) exist along the interval  $[t_0, t_3]$  and the corridor (red) along  $[t_1, t_2]$ . Thus, the buildings were separate at  $t = t_0$ , were then connected by the corridor at  $t = t_1$ , then became disconnected again when the corridor was removed at time  $t_2$ . This configuration remains unchanged until time  $t = t_3$ . The times  $t_0, t_1, t_2$  and  $t_3$  are shown as points along the line representing the front right corner of the connecting corridor.

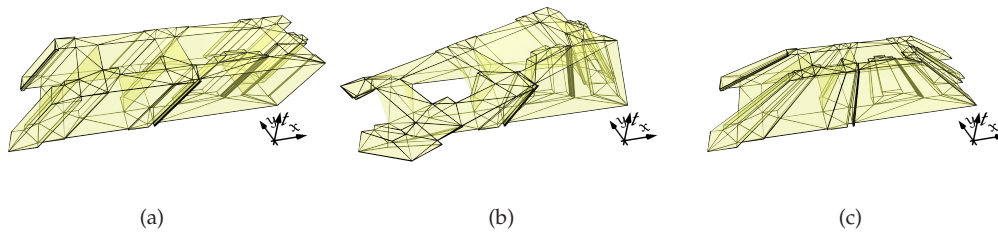


Figure 5: Moving and morphing objects in time can be represented by simple shapes that are related to prisms, such as: (a) parallelipeds for translating objects, (b) twisted prisms for rotating objects, and (c) frustums for scaling objects.

Since many kinds of features in geographic information are represented as planar partitions of polygons, and when considered in time these usually do not overlap spatially or temporally, their resulting 2D space+time representations form 3D space partitions. Such 2D+time representations effectively represent the state of the 2D partition at any point in time. Moreover, their topological relationships in 3D space encode all those contained in 2D space and those in 1D time. For instance, adjacency, and incidence in 2D space are still respectively adjacency and incidence along the first two dimensions, and adjacency and incidence in time are represented by those relationships along the third dimension.

Another interesting aspect to consider is the meaning of a simply connected object in this kind of representation. Starting from a single time interval, which is represented by a simply connected segment of a 1D line, and an area without holes, which is represented by a simply connected region of the plane, a 2D region without holes that has existed for an interval of time is also a simply connected volume in 2D+time. However, not all simply connected volumes in 2D+time represent 2D regions without holes. If a region has a hole during only part of the time interval, it creates a cavity (i.e., a 3D volumetric hole), which in a 3D volumetric representation is still simply connected. However, we should point out that in a true 4D model, simply connected 4D models do not necessarily result in simply connected 3D cross-sections at any given time.

Based on the examples described above, the 4D case is very similar and can be understood by analogy. In 4D, a set of polyhedra in time can be represented as a set of polychora. As a simple example, we can start from a geometrically equivalent configuration as Figure 4(b), but which is now representing 3D space  $xyz$  rather than 2D space+time  $xyt$ . The former corridor footprint now therefore represents a volumetric elevated corridor connecting the second floors of the two buildings.

Figure 6 shows this example using the 3D space+time configuration modeled as 4D space  $xyzt$  by similarly assigning an interval of time during which each represented volume exists. This is done by assuming that the building is constructed from the bottom floors and upwards in three equal increments (i.e.,  $[t_0, t_1]$ ,  $[t_1, t_2]$  and  $[t_2, t_3]$ ), such that the bottom floors exist during the interval  $[t_0, t_3]$ , the middle ones during  $[t_1, t_3]$  and the top ones during  $[t_2, t_3]$ . The 4D model in this figure and the ones which follow are shown using a perspective projection from 4D to 3D, after which we apply another perspective projection down to 2D.

Intuitively, this 4D to 3D projection results in a view where the objects that are farther away are shrunk or scaled inwards, much like the 3D to 2D perspective projection used in most realistic renderings of 3D models and the figures of 3D objects presented in this paper. In short, in order to apply the 4D to 3D perspective projection, we first define a viewpoint with 4D coordinates—which is set as the origin of a new 4D coordinate system—and a viewing direction as a 4D vector—which goes from the viewpoint towards the centroid of the 4D object and is set as the fourth axis of the system. The other axes can be set arbitrarily but must be orthogonal to each other, so we initialize them arbitrarily to  $z$  and  $t$ , and then use a 4D cross product to ensure that they become orthogonal to each other. The final result is then obtained by taking the coordinates of each point in the first three axes and scaling them inwards using the coordinates of each point in the fourth axis (which acts as a measure of the distance from the viewpoint to each point, i.e., the depth). See Arroyo Ohori [10] for more details.

At any one moment in time a polyhedron is still represented as a polyhedron but one that is embedded in 4D space. As the volumes in this example are not moving or changing

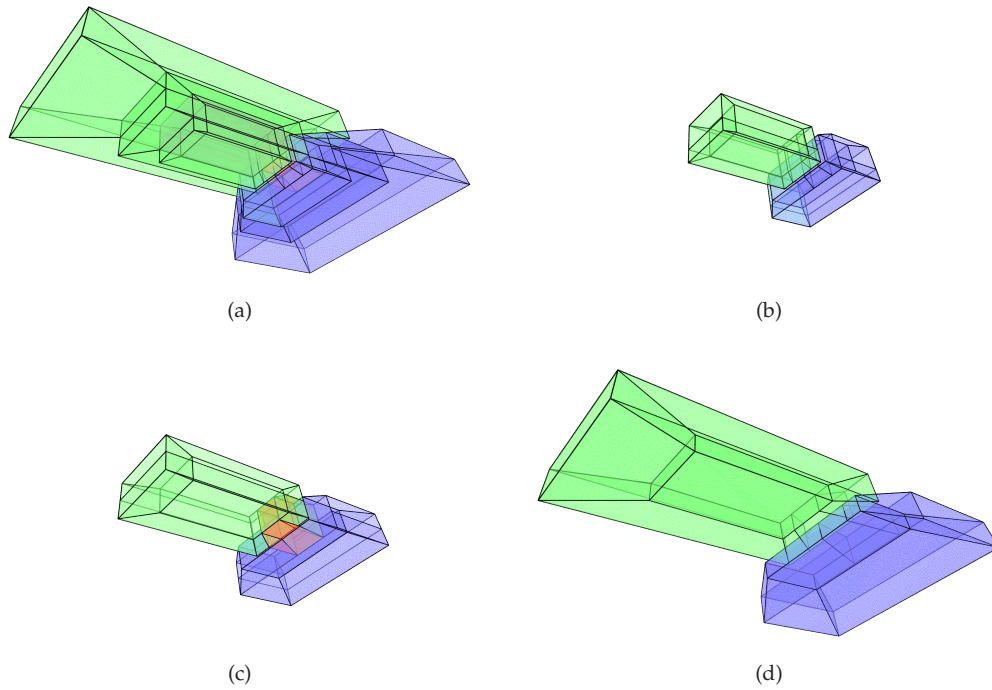


Figure 6: (a) A 3D space  $(x, y, z)$  + time representation of a geometrically equivalent set of volumes as in Figure 4(b) where the volumes of the first floor were assigned the time interval  $[t_0, t_3]$ , those in the second floor  $[t_1, t_3]$  and those in the third floor  $[t_2, t_3]$ , thus representing the construction of a building from the bottom and upwards. The edges and volumes of the 4D model are also shown in three parts for clarity, respectively corresponding to the volumes of the (b) first, (c) second and (d) third floor. The 4-cells of the model, of which there is one per building/corridor in (b), (c) and (d), are bounded by the volumes that are shown.

shape, they take the form of *prismatic polychora*, which are analogous to prisms in 3D and are bounded by identical polyhedral bases whose bounding faces are connected by polyhedral sides.

Like in the 2D+time case, if a set of represented polyhedra form a 3D space partition, the polychora in 3D+time also form a 4D space partition, which describes the state of the polyhedra at any point in time. The topological relationships of adjacency and incidence in 3D+time also represent their analogues in 3D.

Finally, a simply connected object in this kind of 4D representation is similar to the example presented before. If a simply connected volume exists along a single time interval, it can also be represented by a simply connected 4-cell.

### 3.3 Spatiotemporal concepts in a higher-dimensional model

Higher-dimensional models integrating space and time are substantially different from most other types of models presented in Section 2.1. Unlike in other models, where usually only the spatial entities in the model have a natively geometric representation and the temporal and spatiotemporal entities were treated as attributes or as abstract links between other classes, all of the primitives in a higher-dimensional model are spatiotemporal and essentially geometric in nature. In fact, the spatiotemporal primitives and the object equivalences across time in an integrated model directly correspond to point sets in  $\mathbb{R}^4$ , and are thus describable as equations or as specific cell complexes. Notably, these cell complexes can be created, manipulated, and labeled with attributes like all other geometric primitives.

However, most existing spatiotemporal concepts do have clear equivalences in a higher-dimensional model of space and time. Hereafter we therefore map some of the best known spatiotemporal concepts into their geometric equivalents in a higher-dimensional integrated model. Nonetheless, it is good to bear in mind that terminology in different spatiotemporal models differs [106], and so these equivalences might also differ depending on the specific model.

A *moment* or *instant*  $t_0$  corresponds to a hyperplane obtained by an equation of the form  $t = t_0$ , where  $t_0$  is a point along the time axis  $t$ . Such a hyperplane is thus orthogonal to the time axis  $t$ . The hyperplane along all other axes is unbounded and covers the extent  $(-\infty, \infty)$ . Because of this, in terms of an implementation, it is likely easier to store such a hyperplane as a single coordinate along the time dimension. Based on the previous definition, a *snapshot* at a moment  $t_0$  is a 3D cell complex obtained by computing a point-set intersection of the 4D cell complex representing the entirety of the 4D model and the hyperplane defined by the moment  $t_0$ .

A closed *interval* of time  $[t_{\text{start}}, t_{\text{end}}]$  corresponds to a 4D subspace describable by an equation of the form  $t_{\text{start}} \leq t \leq t_{\text{end}}$ . The 4D subspace along all other axes is unbounded. Note that it is similarly possible to describe open intervals, infinite intervals (e. g.,  $[1985, \infty)$ ) and sets of intervals (e. g.,  $[1743, 1945] \cup [2005, \infty)$ ) using similar constructions, all of which have well-defined geometries.

A *location* is a subset of 4D space whose dimension is that of the region that it describes in 3D space plus one. A point in 3D space is thus represented by an infinite line stretching along the time dimension with an extent  $(-\infty, \infty)$  and the same  $xyz$  coordinates as in 3D space. Similarly, a curve thus becomes a subset of an unbounded surface, a surface becomes a subset of an unbounded volume, and a volume becomes a subset of an unbounded 4D subspace.

A 3D *spatial object* or *spatially referenced object* is a 4D cell complex that extends along the time dimension from the creation of the object to its destruction. The geometry of the 3D object at a given time can be obtained by selecting its 4D analogue in the model and computing a point set intersection with a hyperplane defining the moment in time at which the object should be extracted.

An *event* or *process*, interpreted as a point or interval in time that is paired with change in a set of spatial objects or in their attributes, is defined as a cell complex of dimension up to three whose characteristics depend on the kind of event that is being depicted. Admittedly, such a definition is rather vague, but this reflects the great variety among the types of events that can be described. For instance, when such an event describes a change in a 3D object, it can form:

- a 3D cell complex on the common boundary between two or more sets of 4D cell complexes which are modified by the event (in terms of geometry, topology, attributes or the semantics that are implied by being on a certain side of the event); or
- a 3D cell complex on the boundary of one or more 4D cell complexes that are created or modified at the time of the event.

An event might be stored explicitly by marking its corresponding cells as such and attaching them with appropriate semantics about the event. However, much like in other spatiotemporal models, it is also possible to not store all events explicitly and instead deduce certain events from the geometry of the 4D cell complex. For instance, when two 4-cells are adjacent and a significant part of their common bounding volume(s) lie on a hyperplane defined by a moment in time, they are very likely to have been affected (created, destroyed, or have their geometry changed) by an event at that time. In a slightly more complex example, when two 4-cells are not adjacent but nevertheless have some common bounding vertices, edges or faces, they are also likely to have been affected by an event at their common boundary even if their common bounding cells do not lie on a single hyperplane.

In many instances, such 3D cell complexes lie on a hyperplane representing a moment in time, and so the term event seems like a better fit. However, unlike in most spatiotemporal models, there is no reason to limit processes to a single point or interval in time. Processes therefore can comprise a complex series of geometric, topological, and attribute changes occurring in different regions in space and along one or more time points or intervals, and such processes can also be represented implicitly or explicitly.

## 4 Defining useful 4D operations

As shown in Figure 6, even a simple 4D model has a large number of 0-, 1- and 2-cells which visually overlap in any projection down to 2D and 3D, making them difficult to manipulate interactively. Moreover, the 3- and 4-cells in the model are also not appropriate for direct editing as they are rather complex and their bounding facets also obscure each other. Arbitrarily manipulating a 4D model through its up-to-4D primitives and their underlying data structures is thus undoubtedly difficult—not least because of our lack of intuitive understanding of the model’s geometry in dimensions higher than three [43].

However, while defining and using operations based on 0D–4D primitives as input can be quite difficult, for the case of 3D space and time, these kind of operations are not strictly





necessary. The basic manipulation of the model can in fact be performed using a set of cognitively simple operations. This is because many relatively complex 4D spacetime operations in the context of geographic information can be decomposed into separate simpler operations on the three spatial dimensions and on the temporal dimension.

As an example, modeling a moving 3D object can be decomposed into a few easy steps: (i) the 3D object is modeled in standard 3D modeling software, oriented, and placed in its starting location, and exported; (ii) the object is imported into the 4D model at the starting time of the motion (Section 4.1); (iii) the object is selected based on an ID (Section 4.2); (iv) the selection is extruded into a 4D prismatic polychoron up to the end time of the motion (4.3); (v) the 3D base of the extruded object at the end time is selected, which can be accomplished by finding the facet with all vertices using the end time as a fourth coordinate (Section 4.2); and (vi) the base is moved to its final position and orientation in 3D space using a transformation (Section 4.7).

Much like the sequence of operations mentioned above, we propose a scheme where users manipulate 4D models indirectly through operations that either:

- use up-to-4D primitives as input but are already in the system and are conceptually very simple, or
- are more complex but use independent inputs of up-to-3D primitives in 3D space and up-to-1D primitives in time.

The primitives in 3D space can be modeled in standard 3D modeling software, while the simpler 1D temporal primitives, which are intervals or points, can be passed directly to the software. The operations can then be defined and implemented so that they create and manipulate the underlying 4D data structures of the 4D model.

While the number of such operations needed to accommodate all current use cases in 3D GIS and spatiotemporal modeling software as single operations is certainly very large, we have attempted to define a small set of useful 4D operations that can be combined intuitively, can be implemented with relative ease and cover the most common use cases. Many of these correspond directly to typical operations in 2D and 3D geometric modeling [69] and computer graphics [94], including insertion, selection, deletion, and export functions. In addition, we propose using extrusion to convert existing up-to-3D data into up-to-4D data. Finally, we describe a few manipulation operations that change the geometry of a 4D model: generating new snapshots, splitting, and merging geometries, and applying geometric transformations to certain cells in the model. These operations are described briefly below.

## 4.1 Insert 3D model at a given time

The basic insertion operation in a true 4D model should be able to import existing 3D models into the 4D model. For this, the insertion operation takes a set of 0D–3D cells (i.e., a 3D model possibly containing dangling faces, edges, and vertices) and a moment in time (i.e., a point along the temporal dimension or a single timestamp) as input, and importing all the 0-, 1-, 2- and 3-cells in the model at this moment by re-embedding every cell from  $\mathbb{R}^3$  to  $\mathbb{R}^4$ . For this, it is likely best to incrementally construct all the cells using the 4D data structure used in the system, starting from the 0-cells upwards, reusing existing cells in the model where appropriate [12], and performing appropriate preprocessing and validation checks to ensure that the 4D model is valid and self-consistent.

For instance, considering a model with a 4D space partition as a base, it might be necessary to perform a topological reconstruction of the model [12, 35] with a given tolerance threshold or to compute point set intersections between cells. In this manner, lower-dimensional cells that cause other higher-dimensional cells to split (e.g., vertices that fall within an edge) are properly embedded into the same combinatorial structure. Intersecting cells can be similarly split at their common faces.

The attributes of the imported cells can then be individually added to or merged with those of existing cells. For instance, for most dimension-independent data structures, it might be desirable to check that every  $i$ -cell,  $i > 0$ , is properly bounded by a set of  $(i - 1)$ -cells and forms a combinatorial manifold.

Assuming a model with only linear geometries such that the embedding in  $\mathbb{R}^4$  is only defined for the 0-cells, it is possible to use the 1-, 2- and 3-cells in the input 3D model largely as is. For the 0-cells it is however necessary to assign new 4D coordinates to every point. In order to do this, the import function should simply append a given input time  $t_0$  as a fourth coordinate for every point, such that if an input 0-cell is embedded at a point with coordinates  $(x_0, y_0, z_0)$ , its newly assigned 4D coordinates are  $(x_0, y_0, z_0, t_0)$ .

## 4.2 Select cells in the model

In order to further process certain parts of the 4D model, it is necessary to define methods to select certain cells in it. The selection methods used can be similar to those used in 2D/3D GIS, such as selecting using object IDs or particular attribute values (e.g., using algebraic expressions or a set value on the time dimension), object geometry (e.g., using bounding regions of various shapes or simply intervals for every dimension), or object topology (e.g., connectivity to each other or to a given element). In addition, special selections using computed values can be useful, such as selections by a given Lebesgue measure.

However, more complex methods that become more meaningful in higher dimensions are also possible. For instance, selections using object boundaries are useful in order to obtain certain lower-dimensional features, such as a selection involving the cells on the boundary of an already selected object, or a selection of the lower-dimensional cells on the common boundary of two higher-dimensional ones [34]. As another example, a selection based on the objects having a dimension of four can be used to obtain all 3D objects existing along time intervals. Selections based on the computation of topological invariants can be used to find objects with different topological properties, such as using Betti numbers [19] to obtain objects with different numbers of holes of different dimensions (e.g., handles and cavities). Also, a spacetime query verifying whether two 3D objects were ever adjacent is straightforward: if their 4D analogues are adjacent, then at some point in time they were adjacent in 3D as well.

These selection methods can then be used by themselves or combined with each other in order to perform more complex queries. For instance, interior, boundary, and closure operations can be used to refine existing selections, and Boolean set operations can be used to combine multiple selections.

## 4.3 Extrude selected 3D model along a time interval

When a cell of any dimension is inserted into the model using the insertion operation described previously, it is imported into the system as-is, or equivalently, as if it existed only



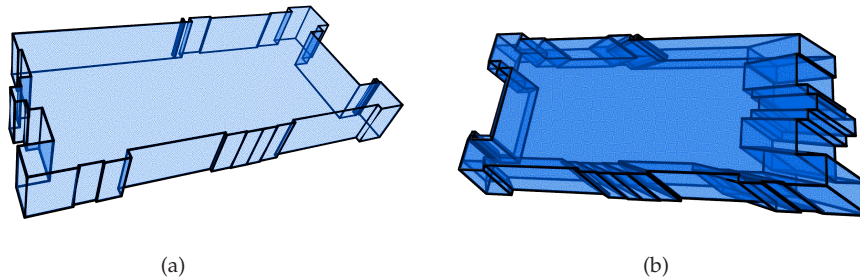


Figure 7: Extruding (a) a set of 0D–3D cells representing a building along a time interval results in (b) a set of prismatic 0D–4D cells.

for a single moment in time. However, many representations of geographic phenomena do not only exist for a moment in time, but instead exist for an interval, which might be closed or open (i.e., containing their endpoints or not), or might be finite (i.e., a line segment along the temporal dimension) or infinite (i.e., a ray or line along the temporal dimension).

Geometrically, an  $i$ -cell existing along a period of time is equivalent to a prismatic  $(i+1)$ -cell—the  $(i+1)$ -dimensional analogue of a prism. Such a prism can be defined intuitively as a cell that is bounded by a set of facets: two identical “top” and “bottom” facets, and a set of other facets that join corresponding ridges of the top and bottom facets.

Higher-dimensional extrusion [13, 42, 67] is a procedure to generate such a prismatic  $(i+1)$ -cell. Given a set of  $i$ -dimensional objects in the form of an  $i$ -dimensional cell complex, and a set of intervals per  $i$ -cell in the complex, it is possible to extrude the cells along the  $(i+1)$ -th dimension, thus creating a  $(i+1)$ -dimensional cell complex. Thus, starting from a set of 0D–3D cells, which are extruded along the time dimension, they are converted into 0D–4D cells. During this time interval the cells remain unchanged (e.g., static, or have the same valid representation). An example of such an operation is shown in Figure 7.

#### 4.4 Delete selected cells

After a set of cells has been selected, the simplest basic operation consists of deleting the selected cells. Such an operation usually only requires finding the primitives associated with the given cells, as well as those for the cells in their boundaries without attributes, and deleting those primitives that are not used in cells that will be preserved in the model.

#### 4.5 Add or remove a snapshot

Another basic operation consists of creating a new snapshot at which all the cells that extend before and after a given time  $t_0$  are split, thus modifying some existing 1D–4D cells and creating an explicit set of new 0D–3D cells at  $t_0$ . This is conceptually equivalent to a method that slices the model with a hyperplane that is orthogonal to the time axis  $t$ . The existing 1D–4D cells extending before and after this time are thus each split into two or more 1D–4D cells. The new 0D–3D cells can then be further processed (e.g., transformed

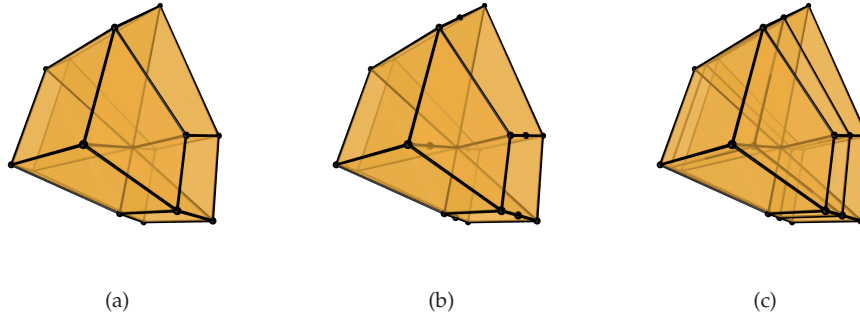


Figure 8: (a) Starting from a tesseract intersected by the hyperplane defined by time  $t_0$ , the procedure to split it at a given time  $t_0$  can be implemented by: (b) splitting all edges that intersect  $t_0$  by generating new vertices, (c) splitting all faces that intersect  $t_0$  by generating new edges, splitting all volumes that intersect  $t_0$  by generating new faces, and splitting all 4-cells that intersect  $t_0$  by generating new volumes. The procedure is shown here using the tesseract  $[-1, 1]^4$ , which is cut along the hyperplane  $t = 0$ .

to model motion) or extracted (e. g., to export the state of a region at a given time as a 3D model).

A relatively simple algorithm that splits the model at  $t_0$  could do this in a few steps in increasing dimension:

1. Split all edges intersecting the hyperplane defined by the moment  $t_0$ . This is done by first finding edges with one endpoint with  $t < t_0$  and the other endpoint with  $t > t_0$ . For every such edge, a new vertex is created at  $t_0$  whose coordinates are a linear combination of the two endpoints of the edge. Finally, the edge is split into two edges using the previous endpoint vertices and the new vertex.
2. Split all faces intersecting the hyperplane  $t_0$ —which have split edges on their boundary—by creating new edges that join the new vertices through the interior of the face only. This can be done using a sweep-plane algorithm on a parallel projection to 2D using a pair of axes where the polygon of the faces does not become degenerate.
3. Split all volumes intersecting the hyperplane  $t_0$ —which have split faces on their boundary—by creating new faces defined by rings of the new edges created in the previous step. Every closed ring defines a new face.
4. Split all 4-cells intersecting the hyperplane  $t_0$ —which have split volumes on their boundary—by creating new volumes that are bounded by the previously created faces. Every set of faces that bound a volume define a new volume, which in the case of dimension-independent data structures based on quasi-cellular manifolds [29, 66] can be found by matching them in pairs at their common boundaries [12].

The first two of these steps are shown using the tesseract (the 4D analogue of a 2D square or 3D cube)  $[-1, 1]^4$  in Figure 8. The last two are not shown since they do not result in appreciably visually different models.



Removing an existing snapshot is much simpler and can be considered as a special case of the merging cells operator described below.

#### 4.6 Split or merge cells

Based on a selection of cells of any dimension existing in the system, a basic modification operation would entail splitting and merging them. In this manner, it is possible to implement many common functions, such as splitting or merging administrative boundaries in 2D, splitting, and merging rooms in a 3D building model (e. g., due to the construction or demolition of a wall), or splitting and merging 4D spatiotemporal instances of those rooms (e. g., due to the creation or deletion of an explicit event).

Merging cells of any dimension is relatively simple, as this operation can be defined based on a selection (e. g., by selecting a cell and then expanding the selection to those adjacent to it), and a set of  $i$ -cells that are connected by adjacency relationships can be merged by removing their common facets. That is, edges can be merged by removing their common bounding vertex, faces can be merged by removing their common bounding edges, volumes by their common bounding faces, and 4-cells by their common bounding volumes.

Splitting cells is somewhat more complex, as this operation invariably needs some kind of geometric input that defines how the cells are split. One option to provide this input is to give a set of cutting hyperplanes, e. g., a 3D space partition and a moment in time, or even a 4D space partition, all of which can be parameterized and passed to a splitting function.

However, for the probably most common way in which such a function would be used, which would split the cells at a given time, it seems significantly easier for a user to export a 3D model of a selection, split edges, faces, and volumes in any 3D modeling software and re-import the resulting 3D model. Depending on the implementation, this procedure might involve first removing the exported cells and then recomputing the topological relationships for the imported cells [12].

The basic constraint for a split or merge operation is that it should be possible to find a 1-to- $n$  (for splitting) or  $n$ -to-1 (for merging) mapping between the input cells and the output cells. A typical example of a workflow using this kind of operation is shown in Figure 9 using a 3D case for clarity, and then a 4D case in Figure 10.

#### 4.7 Apply transformation to selected cells at a given time

As discussed previously in the introduction of this section and in more detail in Arroyo Ohori et al. [15], applying a transformation to either base of a prismatic polychoron (e. g., those generated by extruding any polyhedron) is a simple way to represent a moving object. The unchanged base of the polychoron represents a 3D object at its initial position and orientation, the transformed base represents the 3D object at its final position and orientation, and the 4D subspace in between represents an interpolation of the motion between these two endpoints.

In fact, various transformation matrices can be easily defined to translate, scale, and rotate objects in any dimension [15]. Figure 11 shows the result of applying these transformations to a base of a prismatic polychoron representing a 3D building. Applying similar transformations to only some faces of a given cell can also be used to create other complex

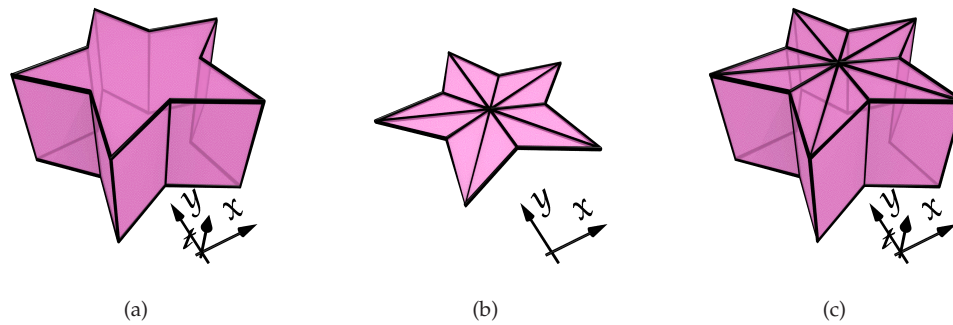


Figure 9: (a) A prism with star bases is generated from importing and extruding a star-shaped polygon. (b) A star-shaped face of the prism is exported and split into a set of triangles. (c) The set of triangles is re-imported and used to substitute the top base of the prism. Note how the side faces connecting corresponding edges of the two bases are maintained, as is the bottom face of the prism.

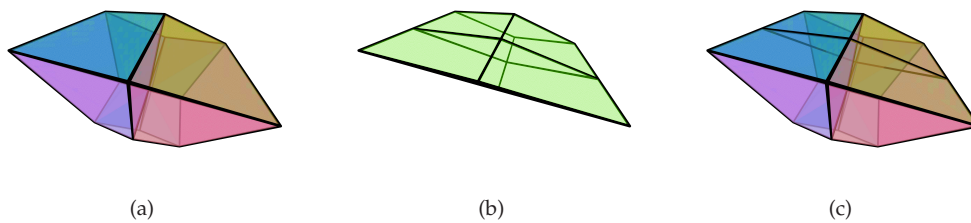


Figure 10: Out of (a) a tesseract, where its 8 bounding cubes are shown in different colors, (b) a single cube is exported and split in half into two cuboids. After (c) the two cuboids are then re-imported and used to substitute the cube, the set of volumes again bound a closed 4-cell.



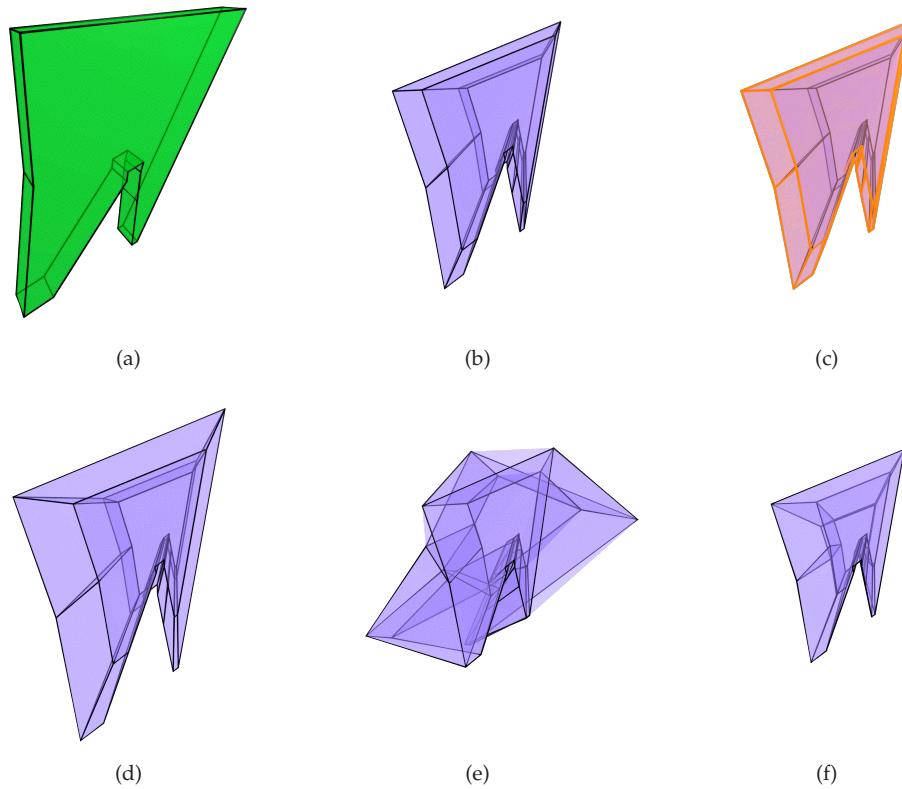


Figure 11: Starting from (a) a polyhedron which is extruded into (b) a prismatic polychoron, 4D transformations can be applied to (c) one of its 3D bases, which is highlighted in orange. The transformations here depict a 3D object that is subject to: (d) translation, (e) rotation, and (f) scaling in time.

objects, such as simply connected 4D objects that are however not simply connected at some snapshots.

However, it is important to note that certain transformations might result in self-intersecting polychora. Among other solutions and depending on the specific case involved, such self-intersections might be resolved in different ways, such as moving vertices, applying incremental extrusion and smaller transformation steps (e. g., for large rotations), or removing degenerate cells from the model [15].

## 4.8 Export 3D model

A final desirable operation, described implicitly before, is exporting a selection of 0D–3D cells from the 4D model as a 3D model. Such a 3D model can then be used for visualization, or used for editing (as in Section 10) and re-imported into the 4D model. In most instances,

the 3D models exported would involve already created snapshots obtained by a selection using the hyperplane defined by a moment in time.

For this, it is necessary to apply a given projection to transform the 4D coordinates of the cells—only the 0-cells in the case of linear geometries—into 3D coordinates. Several types of projections with varying degrees of intuitiveness are possible, such as orthographic projections or the perspective projections used for the figures of this paper, which together are the standard used for visualization in 3D GIS. However, there are other interesting possibilities, such as equiangular projections or to first apply an inwards-outwards projection to a 3-sphere and then a stereographic projection to 3D [15]. Notably, the last approach results in an inwards-outwards time axis in which there are much fewer intersections than in other types of projections, but the resulting models involve rounded volumes, surfaces, and curves or their approximation using a large number of flat volumes, faces, and edges.

## 5 Conclusions and future work

Three-dimensional space and time can be jointly modeled and implemented as four identical dimensions in a 4D model using the principles behind higher-dimensional modeling [10] and dimension-independent data structures [14] such as generalized maps [29,66]. By using a vector modeling approach with up-to-4D objects, it is possible to store even complex situations with no ambiguity, such as objects that move and change in shape at the same time and all the topological relationships between them. While such a representation can also be used in order to model and store lower-dimensional objects (e.g., point clouds and trajectories embedded in 4D), it is not a particularly efficient way to do so. However, using the fourth dimension as a *concept* is still useful for visualization and certain kinds of spatial analyzes, such as the analysis of 3D trajectories in time.

A higher-dimensional spatiotemporal model is substantially different from most other spatiotemporal modeling approaches. Unlike in other models, where usually only the spatial entities in the model have a natively geometric representation and the temporal and spatiotemporal entities were treated as attributes or as abstract links between other classes, all of the primitives in a higher-dimensional model are spatiotemporal and essentially geometric in nature.

While higher-dimensional models and the operations on them can certainly be complex, we believe that they can nevertheless remain intuitive and are implementable in practice. Within this paper, we have focused on a small set of useful 4D operations that are cognitively simple, can be combined intuitively, and cover the most common use cases. Many of these operations are essentially 4D analogues of typical operations in 3D modeling and 3D computer graphics [94]. Some of them also correspond directly to operations in 2D and 3D GIS, including insertion, selection, deletion, and export functions, although special care is needed to re-embed objects from  $\mathbb{R}^3$  to  $\mathbb{R}^4$  and vice versa. In addition,  $n$ D extrusion algorithms can be used to convert 3D objects into 4D ones, where a 4D object represents its 3D analogue remaining static during a specific timeframe. Based on these simplistic extruded 4D models, we also described here a few manipulation operations that change their geometry: generating new snapshots, splitting, and merging 4-cells, and applying geometric transformations to all/part of an object. Combined, these can be used to model many common geographic phenomena, such as implicit and explicit events, histories of administrative subdivisions, moving objects, and simplification/generalization processes.

Overall, we believe that the 4D models obtained by these operations are a good basis for a spatiotemporal GIS, and are significantly more powerful than solutions that focus only on space or time. After all, there is value in a relatively compact set of implementable and cognitively simple operations that can easily produce valid output, especially when testing such validity in dimensions higher than three. In a 4D model, space, and time are equivalent and queries that are space-driven are as easy as those driven by time.

While in this paper we have only discussed the modeling of time in the sense of when events occur (i. e., *real time*), integrated models incorporating other time dimensions are an interesting topic deserving further study. Such models can incorporate more complex time-related features, such as transaction time (i. e., when events are registered into a computer system) and versioning (i. e., different computer models of the same real-world object). Five and higher-dimensional systems can be thus implemented, which are not too different from a theoretical standpoint compared to the 4D system described here. It is however worth noting that defining intuitive 5D and higher operations is significantly more difficult than the 4D operations described in this paper [10].

In the future, we plan to further develop, create relevant algorithms and implement at least some of the operations described in the paper, focusing on the creation and deletion of snapshots (Section 4.5) using arbitrary hyperplanes in  $\mathbb{R}^4$ . Once the CGAL generalized maps package that we are using as a base is released (see acknowledgments below), we will make sure that the developed operations are compatible with it and they will be released as well under an open source license (<http://3d.bk.tudelft.nl/code/>).

## Acknowledgments

We would like to thank Guillaume Damiand for the  $n$ D generalized maps implementation that was used for the examples shown in this paper. It is currently under submission as a potential future package of CGAL<sup>1</sup> (the Computational Geometry Algorithms Library).

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organization for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs (Project code: 11300).

## References

- [1] ABITEBOUL, S., AND HULL, R. Update propagation in the IFO database model. In *Foundations of Data Organization*, S. P. Ghosh, Y. Kambayashi, and K. Tanaka, Eds. Springer, 1987, pp. 319–331. doi:10.1007/978-1-4613-1881-1\_26.
- [2] AIGNER, W., MIKSCH, S., MÜLLER, W., SCHUMANN, H., AND TOMINSKI, C. Visualizing time-oriented data—A systematic view. *Computers & Graphics* 31 (2007), 401–409. doi:10.1016/j.cag.2007.01.030.
- [3] AIGNER, W., MIKSCH, S., MÜLLER, W., SCHUMANN, H., AND TOMINSKI, C. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics* 14, 1 (2008), 47–60. doi:10.1109/TVCG.2007.70415.

---

<sup>1</sup><http://www.cgal.org>

- [4] AKHUNDOV, M. D. *Conceptions of space and time: sources, evolution, directions*. MIT Press, 1986.
- [5] AL-TAHA, K. K., SNODGRASS, R. T., AND SOO, M. D. Bibliography on spatiotemporal databases. *International Journal of Geographical Information Systems* 8, 1 (1994), 95–103. doi:10.1080/02693799408901988.
- [6] ALLEN, J. F., AND FERGUSON, G. Actions and events in interval temporal logic. Tech. Rep. 521, The University of Rochester, 1984.
- [7] ANDRIENKO, G., ANDRIENKO, N., DEMŠAR, U., DRANSCH, D., DYKES, J., FABRIKANT, S., JERN, M., KRAAK, M.-J., SCHUMANN, H., AND TOMINSKI, C. Space, time, and visual analytics. *International Journal of Geographical Information Science* 24, 10 (2010), 1577–1600. doi:10.1080/13658816.2010.508043.
- [8] ANDRIENKO, N., AND ANDRIENKO, G. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer, 2006. doi:10.1007/3-540-31190-4.
- [9] ARMSTRONG, M. P. Temporality in spatial databases. In *Proc. GIS/LIS '88* (1988), American Society for Photogrammetry and Remote Sensing, pp. 880–889.
- [10] ARROYO OHORI, K. *Higher-dimensional modelling of geographic information*. PhD thesis, Delft University of Technology, 2016. doi:10.4233/uuid:39c5b176-398b-4a6f-8578-1dc46e9f37ed.
- [11] ARROYO OHORI, K., BOGUSLAWSKI, P., AND LEDOUX, H. Representing the dual of objects in a four-dimensional GIS. In *Developments in Multidimensional Spatial Data Models*, A. Abdul Rahman, P. Boguslawski, C. Gold, and M. Said, Eds., Lecture Notes in Geoinformation and Cartography. Springer, 2013, pp. 17–31. doi:10.1007/978-3-642-36379-5\_2.
- [12] ARROYO OHORI, K., DAMIAND, G., AND LEDOUX, H. Constructing an  $n$ -dimensional cell complex from a soup of  $(n-1)$ -dimensional faces. In *Proc. First International Conference on Applied Algorithms (ICAA)*, P. Gupta and C. Zaroliagis, Eds., vol. 8321 of *Lecture Notes in Computer Science*. Springer, 2014, pp. 37–48. doi:10.1007/978-3-319-04126-1\_4.
- [13] ARROYO OHORI, K., LEDOUX, H., AND STOTER, J. A dimension-independent extrusion algorithm using generalised maps. *International Journal of Geographical Information Science* 29, 7 (2015), 1166–1186. doi:10.1080/13658816.2015.1010535.
- [14] ARROYO OHORI, K., LEDOUX, H., AND STOTER, J. An evaluation and classification of  $nD$  topological data structures for the representation of objects in a higher-dimensional GIS. *International Journal of Geographical Information Science* 29, 5 (2015), 825–849. doi:10.1080/13658816.2014.999683.
- [15] ARROYO OHORI, K., LEDOUX, H., AND STOTER, J. Defining simple  $nD$  operations based on prismatic  $nD$  objects. In *Joint 3D Athens Conference 2016* (2016), E. Dimopoulou and P. van Oosterom, Eds., vol. IV-2/W1 of *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, ISPRS, pp. 155–162. doi:10.5194/isprs-annals-IV-2-W1-155-2016.

- [16] BACH, B., DRAGICEVIC, P., ARCHAMBAULT, D., HURTER, C., AND CARPENDALE, S. A review of temporal data visualizations based on space-time cube operations. In *Eurographics Conference on Visualization (EuroVis)*, R. Borgo, R. Maciejewski, and I. Viola, Eds. The Eurographics Association, 2014. doi:10.2312/eurovisstar.20141171.
- [17] BACH, B., DRAGICEVIC, P., ARCHAMBAULT, D., HURTER, C., AND CARPENDALE, S. A descriptive framework for temporal data visualizations based on generalized space-time cubes. *Computer Graphics Forum* (2016). doi:10.1111/cgf.12804.
- [18] BASOGLU, U., AND MORRISON, J. The efficient hierarchical data structure for the US historical boundary file. In *Harvard Papers on Geographic Information Systems*, G. Dutton, Ed., vol. 4. Addison-Wesley, 1978.
- [19] BETTI, E. Sopra gli spazi di un numero qualunque di dimensioni. *Annali di Matematica pura ed applicata* 4 (1871), 140–158.
- [20] BIERI, H., AND NEF, W. Elementary set operations with  $d$ -dimensional polyhedra. In *Computational Geometry and its Applications*, H. Noltemeier, Ed., vol. 333 of *Lecture Notes in Computer Science*. Springer, 1988, pp. 97–112. doi:10.1007/3-540-50335-8\_28.
- [21] BLANK, M., GORELICK, L., SHECHTMAN, E., IRANI, M., AND BASRI, R. Actions as space-time shapes. In *Proc. 10th IEEE International Conference on Computer Vision* (2005). doi:10.1109/ICCV.2005.28.
- [22] BRISSON, E. Representing geometric structures in  $d$  dimensions: Topology and order. *Discrete & Computational Geometry* 9 (1993), 387–426. doi:10.1007/BF02189330.
- [23] CHEN, P.-S. The entity-relationship model—Toward a unified view of data. *ACM Transactions on Database Systems* 1, 1 (1976), 9–36. doi:10.1145/320434.320440.
- [24] CHRISMAN, N. R. The role of quality information in the long-term functioning of a geographic information system. *Cartographica* 21, 2 (1984), 79–88.
- [25] CLARAMUNT, C., PARENT, C., SPACCAPIETRA, S., AND THÉRIAULT, M. Database modelling for environmental and land use changes. In *Geographical Information and Planning*, J. C. H. Stillwell, S. Geertman, and S. Openshaw, Eds., Advances in Spatial Science. Springer, 1999, ch. 10, pp. 181–202. doi:10.1007/978-3-662-03954-0\_10.
- [26] CLARAMUNT, C., AND THÉRIAULT, M. Managing time in GIS: An event-oriented approach. In *Proc. International Workshop on Temporal Databases* (1995), J. Clifford and A. Tuzhilin, Eds., pp. 23–42.
- [27] ČOMIĆ, L., AND DE FLORIANI, L. Modeling and Manipulating Cell Complexes in Two, Three and Higher Dimensions, vol. 2 of *Lecture Notes in Computational Vision and Biomechanics*. Springer, 2012, ch. 4, pp. 109–144. doi:10.1007/978-94-007-4174-4\_4.
- [28] COUCLELIS, H. Space, time, geography. In *Geographical Information Systems*, P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, Eds. John Wiley & Sons, 1999, ch. 2, pp. 29–38.
- [29] DAMIAND, G., AND LIENHARDT, P. *Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing*. CRC Press, 2014.

- [30] DE BERG, M., VAN KREVELD, M., OVERMARS, M., AND SCHWARZKOPE, O. *Computational Geometry: Algorithms and Applications*, 3rd ed. SpringerVerlag, 2008. doi:10.1007/978-3-540-77974-2.
- [31] DEMŠAR, U., BUCHIN, K., VAN LOON, E. E., AND SHAMOUN-BARANES, J. Stacked space-time densities: A geovisualisation approach to explore dynamics of space use over time. *GeoInformatica* 19, 1 (2015), 85–115. doi:10.1007/s10707-014-0207-5.
- [32] DEMŠAR, U., AND ÇÖLTEKIN, A. Quantifying the interactions between eye and mouse movements on spatial visual interfaces through trajectory visualisations. In *Proc. Workshop on Analysis of Movement Data at GIScience 2014* (2014). doi:10.5167/uzh-105671.
- [33] DESCARTES, R. *Discours de la méthode*. Jan Maire, Leyde, 1637.
- [34] DIAKITÉ, A. A. *Application des cartes combinatoires à la modélisation géométrique et sémantique des bâtiments*. PhD thesis, Université Claude Bernard - Lyon I, 2015.
- [35] DIAKITÉ, A. A., DAMIAND, G., AND VAN MAERCKE, D. Topological reconstruction of complex 3D buildings and automatic extraction of levels of detail. In *Proc. 2nd Eurographics Workshop on Urban Data Modelling and Visualisation* (2014). doi:10.2312/udmv.20141074.
- [36] DOWNS, J. A., HORNER, M. W., HYZER, G., LAMB, D., AND LORAAMM, R. Voxel-based probabilistic space-time prisms for analysing animal movements and habitat use. *International Journal of Geographical Information Science* 28, 5 (2014), 875–890. doi:10.1080/13658816.2013.850170.
- [37] DUBE, M. P., AND EGENHOFER, M. J. An ordering of convex topological relations. In *Geographic Information Science*, N. Xiao, M.-P. Kwan, M. F. Goodchild, and S. Shekhar, Eds., vol. 7478 of *Lecture Notes in Computer Science*. Springer, 2012, pp. 72–86. doi:10.1007/978-3-642-33024-7\_6.
- [38] EARMAN, J. How to talk about the topology of time. *Noûs* 11, 3 (1977), 211–226. doi:10.2307/2214763.
- [39] EGENHOFER, M. J., AND FRANZOSA, R. D. Point-set topological spatial relations. *International Journal of Geographical Information Systems* 5, 2 (1991), 161–174. doi:10.1080/02693799108927841.
- [40] ERWIG, M., GÜTING, R. H., SCHNEIDER, M., AND VAZIRGIANNIS, M. Spatio-temporal data types: An approach to modeling and querying moving objects in databases. *GeoInformatica* 3, 3 (1999), 269–296. doi:10.1023/A:1009805532638.
- [41] ERWIG, M., AND SCHNEIDER, M. Spatio-temporal predicates. *IEEE Transactions on Knowledge and Data Engineering* 14, 4 (2002), 881–901. doi:10.1109/TKDE.2002.1019220.
- [42] FERRUCCI, V. Generalised extrusion of polyhedra. In *Proc. 2nd ACM Solid Modelling*. ACM, 1993, pp. 35–42. doi:10.1145/164360.164376.



- [43] FRANK, A. U. Four-dimensional representation in human cognition and difficulties with demonstrations: A commentary on Wang. *Spatial Cognition & Computation* 14 (2014), 114–120. doi:10.1080/13875868.2014.885526.
- [44] GALTON, A. Fields and objects in space, time, and space-time. *Spatial Cognition and Computation* 4, 1 (2004), 39–68. doi:10.1207/s15427633scc0401\_4.
- [45] GOLD, C. M. Data structures for dynamic and multidimensional GIS. In *Proc. 4th ISPRS Workshop on Dynamic and Multi-dimensional GIS* (2005), pp. 36–41.
- [46] GOLD, C. M., NANTEL, J., AND YANG, W. Outside-in: an alternative approach to forest map digitizing. *International Journal of Geographical Information Systems* 10, 3 (1996), 291–310. doi:10.1080/02693799608902080.
- [47] GUDMUNDSSON, J., LAUBE, P., AND WOLLE, T. Movement patterns in spatio-temporal data. In *Encyclopedia of GIS*. Springer, 2008, pp. 726–732. doi:10.1007/978-0-387-35973-1\_823.
- [48] GÜTING, R. H., BÖHLEN, M. H., ERWIG, M., JENSEN, C. S., LORENTZOS, N. A., SCHNEIDER, M., AND VAZIRGIANNIS, M. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems* 25, 1 (2000), 1–42. doi:10.1145/352958.352963.
- [49] HÄGERSTRAND, T. What about people in regional science? *Papers of the Regional Science Association* 24, 1 (1970), 6–21. doi:10.1111/j.1435-5597.1970.tb01464.x.
- [50] HALLOT, P., AND BILLEN, R. Generalized life and motion configurations reasoning model. In *International Workshop on Moving Objects: From Natural to Formal Language* (2008). doi:10.1007/978-3-540-87991-6-39.
- [51] HAMRE, T., MUGHAL, K. A., AND JACOB, A. A 4D marine data model: Design and application in ice monitoring. *Marine Geodesy* 20, 2–3 (1997), 121–136. doi:10.1080/01490419709388100.
- [52] HATCHER, A. *Algebraic Topology*. Cambridge University Press, 2002.
- [53] HORNE, J. S., GARTON, E. O., KRONE, S. M., AND LEWIS, J. S. Analyzing animal movements using Brownian bridges. *Ecology* 88, 9 (2007), 2354–2363. doi:10.1890/06-0957.1.
- [54] HORNSBY, K., AND EGENHOFER, M. J. Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence* 36, 1–2 (2002). doi:10.1023/A:1015812206586.
- [55] HUNTER, G. J., AND WILLIAMSON, I. P. The development of a historical cadastral database. *International Journal of Geographical Information Systems* 4, 2 (1990), 169–179. doi:10.1080/02693799008941538.
- [56] ISO. *ISO19107 Geographic Information — Spatial Schema*. International Organization for Standardization, 2005.
- [57] ISO. *ISO19108 Geographic information — Temporal schema*. International Organization for Standardization, 2005.

- [58] JIANG, J., AND WORBOYS, M. Event-based topology for dynamic planar areal objects. *International Journal of Geographical Information Science* 23, 1 (2009), 33–60. doi:10.1080/13658810802577247.
- [59] KOWALSKI, R., AND SERGOT, M. A logic-based calculus of events. *New Generation Computing* 4 (1986), 67–95. doi:10.1007/BF03037383.
- [60] KRAAK, M.-J. The space-time cube revisited from a geovisualization perspective. In *Proc. 21st International Cartographic Conference* (2003), pp. 1988–1996.
- [61] LANGRAN, G. Producing answers to spatial questions. In *Auto-Carto 10: Technical Papers of the 1991 ACSM-ASPRS Annual Convention* (1991).
- [62] LANGRAN, G., AND CHRISMAN, N. R. A framework for temporal geographic informations. *Cartographica* 25, 3 (1988), 1–14.
- [63] LATTUADA, R. *A triangulation based approach to three dimensional geoscientific modelling*. PhD thesis, Department of Geography, Birkbeck College, University of London, UK, 1998.
- [64] LE, H. H. Spatio-temporal data construction. *ISPRS International Journal of Geo-Information* 2 (2013), 837–853. doi:10.3390/ijgi2030837.
- [65] LEDOUX, H., AND GOLD, C. M. Modelling three-dimensional geoscientific fields with the Voronoi diagram and its dual. *International Journal of Geographical Information Science* 22, 5 (2008), 547–574. doi:10.1080/13658810701517120.
- [66] LIENHARDT, P. *n*-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications* 4, 3 (1994), 275–324. doi:10.1142/S0218195994000173.
- [67] LIENHARDT, P., SKAPIN, X., AND BERGEY, A. Cartesian product of simplicial and cellular structures. *International Journal of Computational Geometry and Applications* 14, 3 (2004), 115–159. doi:10.1142/S0218195904001408.
- [68] LIGOZAT, G. Towards a general characterization of conceptual neighborhoods in temporal and spatial reasoning. In *Proc. AAAI Workshop on Spatial and Temporal Reasoning* (1994), pp. 745–750.
- [69] MÄNTYLÄ, M. *An introduction to solid modeling*. Computer Science Press, New York, USA, 1988.
- [70] MASON, N. C., O’CONNELL, M. A., AND BELL, S. B. M. Handling four-dimensional geo-referenced data in environmental GIS. *International Journal of Geographical Information Systems* 8, 2 (1994), 191–215. doi:10.1080/02693799408901994.
- [71] MASUDA, H. Topological operators and Boolean operations for complex-based non-manifold geometric models. *Computer-Aided Design* 25, 2 (1993). doi:10.1016/0010-4485(93)90097-8.
- [72] MCCARTHY, J., AND HAYES, P. J. Some philosophical problems from the standpoint of artificial intelligence. In *Reading in Artificial Intelligence*. Tioga, 1969.



- [73] MCKENZIE, J. W., WILLIAMSON, I. P., AND HAZELTON, N. 4-D adaptive GIS: Justification and methodologies. Tech. rep., Department of Geomatics, The University of Melbourne, 2001.
- [74] MILLER, H. J. Modelling accessibility using space-time prism concepts within geographical information systems. *International Journal of Geographical Information Systems* 5, 3 (1991), 287–301. doi:10.1080/02693799108927856.
- [75] MINKOWSKI, H. Die Grundgleichungen für die elektromagnetischen Vorgänge in bewegten Körpern. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* (1908), 53–111.
- [76] O’CONAILL, M. A., BELL, S. B. M., AND MASON, N. C. Developing a prototype 4D GIS on a transputer array. *ITC Journal* 1 (1992), 47–54.
- [77] PARENT, C., SPACCAPIETRA, S., AND ZIMÁNYI, E. Spatio-temporal conceptual models: Data structures + space + time. In *Proc. 7th ACM Symposium on Advances in Geographic Information Systems* (1999). doi:10.1145/320134.320142.
- [78] PELEKIS, N., THEODOULIDIS, B., KOPANAKIS, I., AND THEODORIDIS, Y. Literature review of spatio-temporal database models. *The Knowledge Engineering Review* 19, 3 (2004), 235–274. doi:10.1017/S02698890400013X.
- [79] PEUCKER, T. K., AND CHRISMAN, N. R. Cartographic data structures. *The American Cartographer* 2, 1 (1975), 55–69. doi:10.1145/965143.563306.
- [80] PEUQUET, D. J. It’s about time: A conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers* 84, 3 (1994), 441–461. doi:10.1111/j.1467-8306.1994.tb01869.x.
- [81] PEUQUET, D. J. Making space for time: Issues in space-time data representation. *GeoInformatica* 5, 1 (2001), 11–32. doi:10.1023/A:1011455820644.
- [82] PEUQUET, D. J., AND DUAN, N. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Science* 9, 1 (1995), 7–24. doi:10.1080/02693799508902022.
- [83] PIGOT, S., AND HAZELTON, N. W. J. The fundamentals of a topological model for a four-dimensional GIS. In *Proc. 5th International Symposium on Spatial Data Handling* (1992), pp. 580–591.
- [84] POINCARÉ, M. Analysis situs. *Journal de l’École polytechnique* 2, 1 (1895), 1–123.
- [85] PRICE, R., TRYFONA, N., AND JENSEN, C. S. Extended spatiotemporal uml: Motivations, requirements and constructs. *Journal of Database Management* 11, 4 (2000). doi:10.4018/jdm.2000100102.
- [86] PRIOR, A. N. *Time and Modality*. Clarendon Press, 1957.
- [87] QIANG, Y., CHAVOSHI, S. H., LOGGHE, S., DE MAEYER, P., AND VAN DE WEGHE, N. Multi-scale analysis of linear data in a two-dimensional space. *Information Visualization* 13, 3 (2014), 248–265. doi:10.1177/1473871613477853.

- [88] RAPER, J. *Multidimensional geographic information science*. CRC Press, 2000.
- [89] RENOLEN, A. History graphs: Conceptual modeling of spatio-temporal data. *GIS Frontiers in Business and Science* 2 (1996).
- [90] RENOLEN, A. Modelling spatiotemporal information: The spatiotemporal object model. Tech. rep., Norwegian University of Science and Technology, 1997.
- [91] RENOLEN, A. *Concepts and Methods for Modelling Temporal and Spatiotemporal Information*. PhD thesis, Norwegian University of Science and Technology, 1999.
- [92] RIEMANN, B. *Ueber die Hypothesen, welche der Geometrie zu Grunde liegen*. PhD thesis, Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen, 1868.
- [93] ROSSIGNAC, J., AND O'CONNOR, M. SGC: A dimension-independent model for pointsets with internal structures and incomplete boundaries. In *Proc. IFIP Workshop on CAD/CAM* (1989), M. Wosny, J. Turner, and K. Preiss, Eds., pp. 145–180.
- [94] SCHNEIDER, P. J., AND EBERLY, D. H. *Geometric Tools for Computer Graphics*. Morgan Kaufmann, 2003.
- [95] SHEWCHUK, J. R. General-dimensional constrained Delaunay and constrained regular triangulations, I: Combinatorial properties. *Discrete & Computational Geometry* 39, 1003 (2008), 580–637. doi:10.1007/s00454-008-9060-3.
- [96] SOHANPANA, C. Extension of a boundary representation technique for the description of  $n$  dimensional polytopes. *Computers & Graphics* 13, 1 (1989), 17–23. doi:10.1016/0097-8493(89)90032-0.
- [97] STORY, P. A., AND WORBOYS, M. F. A design support environment for spatio-temporal database applications. In *Spatial Information Theory: A Theoretical Basis for GIS* (1995), A. U. Frank and W. Kuhn, Eds., vol. 988 of *Lecture Notes in Computer Science*, Springer, pp. 413–430. doi:10.1007/3-540-60392-1\_27.
- [98] THEODORIDIS, Y., SILVA, J. R. O., AND NASCIMENTO, M. A. On the generation of spatiotemporal datasets. Tech. rep., TimeCenter, 1999.
- [99] TRYFONA, N., AND JENSEN, C. S. Conceptual data modeling for spatiotemporal applications. *GeoInformatica* 3, 3 (1999), 245–268. doi:10.1023/A:1009801415799.
- [100] VAN DE WEGHE, N., DE ROO, B., QIANG, Y., VERSICHELE, M., NEUTENS, T., AND DE MAEYER, P. The continuous spatio-temporal model (CSTM) as an exhaustive framework for multi-scale spatio-temporal analysis. *International Journal of Geographical Information Science* 28, 5 (2014), 1047–1060. doi:10.1080/13658816.2014.886329.
- [101] VAN OOSTEROM, P. Maintaining consistent topology including historical data in a large spatial database. In *Proc. Auto-Carto 13* (1997).
- [102] VAN OOSTEROM, P., AND STOTER, J. 5D data modelling: Full integration of 2D/3D space, time and scale dimensions. In *Proc. 6th International Conference on GIScience*, S. I. Fabrikant, T. Reichenbacher, M. van Kreveld, and C. Schlieder, Eds. Springer, 2010, pp. 311–324. doi:10.1007/978-3-642-15300-6\_22.



- [103] VARMA, H., BOUDREAU, H., AND PRIME, W. A data structure for spatio-temporal databases. In *Proc. IHO Review* (1990), pp. 1–10.
- [104] WACHOWICZ, M., AND HEALY, R. G. Towards temporality in GIS. In *Innovations in GIS*. Taylor & Francis, 1994.
- [105] WORBOYS, M. A model for spatio-temporal information. In *Proc. 5th International Symposium on Spatial Data Handling* (1992), pp. 602–611.
- [106] WORBOYS, M. Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science* 19, 1 (2005), 1–28. doi:10.1080/13658810412331280167.
- [107] WORBOYS, M. F. A generic model for planar geographical objects. *International Journal of Geographical Information Systems* 6, 5 (1992), 353–372. doi:10.1080/02693799208901920.
- [108] WORBOYS, M. F. A unified model for spatial and temporal information. *The Computer Journal* 37, 1 (1994), 26–34. doi:10.1093/comjnl/37.1.26.
- [109] WORBOYS, M. F., HEARNshaw, H. M., AND MAGUIRE, D. J. Object-oriented data modelling for spatial databases. *International Journal of Geographical Information Systems* 4, 4 (1990), 369–383. doi:10.1080/02693799008941553.
- [110] YUAN, M. Wildfire conceptual modeling for building GIS space-time models. In *Proc. GIS/LIS'94* (1994), pp. 860–869.