The Honeycomb Model of Spatio-Temporal Partitions*

Martin Erwig & Markus Schneider

FernUniversität Hagen, Praktische Informatik IV
58084 Hagen, Germany
[erwig|markus.schneider]@fernuni-hagen.de

Abstract. We define a formal model of spatio-temporal partitions which can be used to model temporally changing maps. We investigate new applications and generalizations of operations that are well-known for static spatial maps. We then define a small set of operations on spatio-temporal partitions that are powerful enough to express all these tasks and more. Spatio-temporal partitions combine the general notion of temporal objects and the powerful spatial partition abstraction into a new, highly expressive spatio-temporal data modeling tool.

1 Introduction

The subject of this paper is the *temporal evolution of maps*. The metaphor of a *map* has turned out to be a fundamental and ubiquitous spatial concept in many spatially-oriented disciplines like geography and cartography as well as in computer-assisted systems like geographical informations systems (GIS), spatial database systems, and image database systems, but also simply for human's spatial orientation in everyday life. A map is a widely recognized geometric structure that is capable of carrying a large amount of information and that can be well displayed in visual form.

The central elements of maps are so-called *partitions* whose importance is already reflected by the fact that the notion "map" is frequently used as a synonym for "partition". The mathematical understanding of a partition differs slightly (but decisively) from its spatial interpretation. In [6], we have motivated and formally defined the notion of *spatial partition*. Spatial partitions are proposed as a generic data type that can be used to model arbitrary maps and to support spatial analysis tasks. Three fundamental, powerful operations on partitions have been identified that allow one to express and even to generalize all known application-specific operations on maps. Examples of spatial partitions are the subdivision of the world map into countries, classification of rural areas according to their agricultural use, areas of different degrees of air pollution, distribution of ethnic groups, areas with different spoken languages, etc.

* This research was partially supported by the CHOROCHRONOS project, funded by the EU under the Training and Mobility of Researchers Programme, Contract No. ERB FMRX-CT96-0056.

A spatial partition is a subdivision of the plane into pairwise disjoint *regions* where regions are separated from each other by *boundaries* and where each region is associated with an attribute having simple or even complex structure. That is, a region (possibly composed of several components) with an attribute incorporates all points of a spatial partition having this attribute. A spatial partition implicitly models topological relationships between the participating regions which can be regarded as integrity constraints. First, it expresses neighborhood relationships for different regions that have common boundaries. This property is immediately visible on a map. A second related aspect is that different regions of a partition are always disjoint (if we neglect common boundaries) so that a visual representation of a partition has a very simple structure and is easy to grasp. Both topological properties of spatial partitions will be denoted as *partition constraints*.

As a purely geometric structure, a map yields only a *static* description of spatial entities and required constraints between them. Recently, strong research efforts have been made in spatial and temporal data modeling to integrate both directions into a common research branch, called *spatio-temporal modeling*, and to construct *spatio-temporal databases*. The central aim is to observe and to model evolutions of spatial phenomena over time. Spatio-temporal data models for single, self-contained entities like *moving points* or *evolving regions* have already been studied in [4, 5]. But so far, data models for spatial partitions changing over time have not been considered.

The main objectives of this paper are twofold. First, we give some examples of interesting application scenarios for spatial partitions evolving over time and illustrate their characteristic features. Using a *three-dimensional (3D) view*, we will see that temporally changing spatial partitions can be well visualized by *three-dimensional spatial partitions* where each temporally changing region develops into a volume. Partition constraints adapted to the 3D case are maintained by the volumes. Additionally, application-specific operations on two-dimensional partitions can be "lifted" to the spatio-temporal case. The interesting aspect here is that these operations can be even further generalized. We will demonstrate this by examples.

Second, we study the temporal development of spatial partitions thoroughly and give a formal semantics to them and to their operations. This leads us to so-called *spatio-temporal partitions*, that is, to collections of regions satisfying the partition constraints for each time of their lifespan and maintaining these constraints over time. Temporal changes of spatial partitions can occur either continuously (for example, distribution of air pollution or temperature zones) or in discrete and stepwise constant steps (for example, reunification of West and East Germany, splitting of Yugoslavia). If we imagine a spatio-temporal partition and a time axis perpendicular to the Euclidean plane, for each time slice parallel to the xy-axis, we obtain a stationary, two-dimensional spatial partition which changes over time due to altering shapes, sizes, or attribute values of regions. This imagination corresponds to the *temporal object view* already described in [8]; it is based on the observation that everything that changes over

time can be considered as a function over time. In the context here, spatio-temporal partitions can be viewed as functions from time to a two-dimensional spatial partition.

The remainder of the paper is structured as follows. Section 2 recalls related work with respect to spatial partitions and spatio-temporal objects. Section 3 informally describes the structure of and operations on spatio-temporal partitions from an application point of view. In Section 4 our formal model of spatial partitions is briefly reviewed which serves as the foundation for a formal model of spatio-temporal partitions which is then given in Section 5. Additionally, some more operations are introduced emphasizing the temporal aspect of spatio-temporal partitions. Section 6 draws some conclusions.

2 Related Work

In this section we will briefly review related work on spatial partitions and on spatio-temporal data modeling as far as it is relevant for this paper. We are not aware of any work on the combination of both research aspects.

2.1 Spatial Partitions

Spatial partitions or maps have been identified as a basic *spatial concept* to organize and conceptualize our perception and understanding of space. They correspond to humans' cognitive experience and knowledge of areal phenomena in the real world. If we consider the same space with respect to two different thematic or cognitive aspects (for example, districts and cereals) modeled as two partitions, their overlay is a partition again.

Maps arising from classifying space according to some aspect are frequently called *thematic maps* or *categorical coverages* [10, 21]. But these concepts mainly focus on partitions of attribute values alone – spatial operations are completely ignored. In particular, *boundaries* are not considered which play an important role in connection with geometric intersection.

In geographic applications and systems spatial partitions are regarded as a fundamental and user-friendly data modeling tool offering a powerful basis for coping with spatial analysis and cartographical tasks [1, 9, 10, 14, 20, 21]. Distinct features over the same space can be combined and evaluated under different requirements. Partition-based spatial analysis functions include operations like overlay, generalization, and reclassification. They all produce a new partition as a result.

From a data type perspective there have been some unsatisfactory proposals in the past to model partitions. In [12] a spatial data type *area* is suggested to model the partition constraints. Within the framework of an extended relational data model the set of polygons occurring in a relation as a column of an attribute of type *area* has to fulfill the integrity constraint that all polygons are adjacent or disjoint to each other. Unfortunately, the maintenance of this property is not supported by the data model, rather it is up to the user's responsibility. A generic

data type for partitions, called *tessellation*, is informally introduced in [14] as a specialized type for sets of polygons; this type can be parametrized with an attribute of a yet unspecified type. In [13] so-called *restriction types* have been proposed. This concept allows one to restrict the general type for regions to subtypes whose values all satisfy a specific topological predicate (like *disjoint*) and which nevertheless inherit the properties and operations of the more general type for regions.

A rigorous and thorough formal definition of spatial partitions and of application-specific operations defined on them is given in [6]. The basic idea is that a spatial partition is a mapping from the Euclidean space $\mathbb{R}^2$ to some *label type*, that is, regions of a partition are assigned single labels (see Section 4.1). Adjacent regions have different labels in their interior, and a boundary is assigned the pair of labels of both adjacent regions.

In [6] all application-specific operations have been reduced to the three fundamental and powerful operations *intersection*, *relabel*, and *refine* (see Section 4.2). Intersecting two spatial partitions means to compute the geometric intersection of all regions and to produce a new spatial partition; each resulting region is labeled with the pair of labels of the original two intersecting regions, and the values on the boundaries are derived from these. Relabeling a spatial partition has the effect of changing the labels of its regions. This can happen by simply renaming the label of each region. Or, in particular, distinct labels of two or more regions are mapped to the same new label. If some of these regions are adjacent in the partition, the border between them disappears, and the regions are fused in the result partition. Relabeling has then a coarsening effect. Refining a partition means to look with a finer granularity on regions and to reveal and to enumerate the connected components of regions.

All of the following application-specific operations are covered by these three operations:

- *Overlay.* This most important application operation on maps [1, 9, 13, 12, 14–17, 20] allows to lay two partitions with different attribute categories on top of each other and to combine them through geometric intersection into a new partition of disjoint and adjacent regions. The attributes from the input partitions are then either distributed to each region of the result partition or appropriately mapped to a new attribute. The underlying basic operation is obviously intersection.
- *Reclassify* [1, 14] retains the geometric structure of the spatial partition and transforms all or some partition attributes to new or modified attributes. It is a special case of relabeling.
- *Fusion* [13–17] is a kind of grouping operation with subsequent geometric union. It merges neighbored regions of a partition with respect to partially identical attributes and is also a special case of relabeling.
- *Cover* [17] forms the geometric union of all regions of a partition and yields a result partition consisting of one region. Since *cover* is a special case of *fusion*, it can again be realized by relabeling.
- *Clipping* [17] computes the intersection of a partition and a given rectangular window. As a special case of *overlay*, it can be expressed by intersection. We

have generalized this operation and allow general so-called "unit partitions" as clipping windows [6].
- *Difference* [14] takes two spatial partitions defined over the same attribute domain and computes the geometric difference of their point sets. All the regions of the first partition are maintained in the result partition except for those parts that have the same attributes in both partitions. We have generalized this operation in several ways; it can be reduced to a combination of intersecting and relabeling [6].
- *Superimposition* [17] allows to superimpose the regions of a partition onto another partition and to cover and erase parts of the other partition. It is a special case of intersection.
- *Window* [17] retrieves those complete regions of a spatial partition whose intersection with a given window is not empty. Its definition, which was generalized in [6], is based on all three fundamental operations. In particular, this is the only operation that requires the partition operation *refine*.

2.2 Spatio-Temporal Objects

So far, only a few data models for spatio-temporal data have been proposed. They all focus on describing the temporal development of single, self-contained spatial objects but do not take into account collections of evolving spatial objects possibly satisfying some constraints over time. Either spatial data models [22] or temporal data models [11, 2] have been extended to become spatio-temporal. The main drawback of all these approaches is that they are incapable of modeling *continuous* changes of spatial objects over time.

Our approach to dealing with spatio-temporal data supports an integrated view of space and time and incorporates the treatment of continuous spatial changes. It is based on the concept of *spatio-temporal data types* [4, 5]. These data types are designed as abstract data types whose values can be integrated as complex entities into databases and whose definition and integration into databases is independent of a particular DBMS data model.

A temporal version of an object of type α is given by a function from *time* to α. Spatio-temporal objects like *moving points* or *evolving regions* are regarded as special instances of temporal objects where α is a spatial data type like *point* or *region*. A moving point describes a point changing its location in the Euclidean plane over time. An evolving region is a temporally changing region that can move and/or grow/shrink. It can even disappear or reappear, and its components can either split or merge.

A straightforward and very instructive view of spatio-temporal objects is to conceptualize and visualize their temporal evolution as purely geometric, *three-dimensional* objects, that is, the time axis is regarded as a third geometric dimension. An evolving region is then represented as a *volume* in 3D space (with z-monotonic surfaces), and a moving point is then visualized as a (z-monotonic) *3D curve*. Any intersection parallel to the xy-plane yields a spatial object, that is, a region or a point. These two views, the *temporal object view* and the *three-dimensional object view*, together with the concept of 2D spatial

partitions described in Section 2.1 will serve as the main design guidelines for modeling spatio-temporal partitions.

Note that in the same way as spatial partitions cannot be modeled adequately with a type of spatial regions, evolving regions are insufficient to capture the inherent constraints of spatio-temporal partitions.

3 Applications of Spatio-Temporal Partitions

Spatio-temporal partitions or "temporal maps" have a wide range of interesting applications. In this section we will have a look at some of these applications and demonstrate the essence and power of spatio-temporal partitions and operations defined on them. Section 3.1 introduces selected applications for temporal maps and uses them to explain the structure of spatio-temporal partitions. Section 3.2 briefly deals with a possible visualization of spatio-temporal partitions in user interfaces. Finally, Section 3.3 considers applications that combine spatio-temporal partitions through operations.

3.1 Structure of Spatio-Temporal Partitions

The following examples comprise time-dependent spatial mapping and analysis tasks as they are relevant for cartography, GIS, and other spatially-related application areas. Their actual power is later revealed when two partitions are combined appropriately (see Section 3.3).

From an application point of view, we can generally distinguish two categories of temporal maps. The first category incorporates applications whose temporal changes are discrete. For example, consider the temporal development of any hierarchical decomposition of space into administrative or cadastral units like the world map into countries or districts into land parcels. Another application is the classification of rural areas according to their agricultural use (like the cultivation of cereals) over time. A further example is a chronology of ruling parties in countries.

A characteristic feature of applications of this first category is that the number of discrete temporal changes is finite and that there is no change between any two subsequent *temporal change points*, that is, the development is stepwise constant which is a special form of (semi-)continuity. For each time between two temporal change points we expect and obtain a unique and correct spatial partition.

The open issue now is what happens at temporal change points with their abrupt transition from one spatial partition to another. If we consider the time point when West and East Germany were reunified, did the spatial partition before or after the reunification belong to this time point? Since we cannot come to an objective decision but only know that not both spatial partitions can simultaneously belong to the temporal change point, we have to decide arbitrarily and to assign one of both spatial partitions to it. This, in particular, maintains the functional character of our temporal object view. We have chosen

to ascribe the temporally later spatial partition to a temporal change point. Mathematically this means that we permit a finite set of time points where the temporal function is not continuous.

The examples reveal that after a temporal change point the continuity of the temporal function proceeds for some time interval up to the next temporal change point; there are no "thin, isolated slices" containing single spatial partitions at temporal change points. Consider the result of the reunification of West and East Germany which after that event has lasted up to now. Hence, we have to tighten our requirement in the sense that mathematically the temporal function has to be (*upper*) *semicontinuous* at each time. Intuitively, this means that the temporal function has to be continuous from the upper side.

The second category includes applications whose temporal changes are continuous or smooth. Consider the temporal evolution of climatic phenomena like temperature zones or high/low pressure areas, areas of air pollution with distinct degrees of intensity, or developments of forest fires in space and time. They all show a very dynamic and attribute-varying behavior over time. Application examples which have by far slower temporal changes are the increasing spread of ethnic or religious groups, the decreasing extent of mineral resources like oil fields during the course of time due to exploitation, or the subdivision of space into areas with different sets of spoken languages over time.

So far, we have intuitively described the *temporal object view* of spatio-temporal partitions. An alternative imagination of spatio-temporal partitions is given by the *three-dimensional object view*. The idea is to regard the time axis as a third geometric dimension, the z-axis, and to represent the temporal evolution of regions of a spatial partition as solid 3D volumes. This leads to *three-dimensional partitions* where any two volumes are either disjoint, or they are adjacent and have common boundary parts. The predicates "disjoint" and "adjacent" denote topological relationships in three-dimensional space. We will denote these 3D volumes as *partition volumes*.

An interesting observation is that partition volumes cannot be shaped arbitrarily. They must reflect the functional character of spatio-temporal partitions. That implies that the boundaries with respect to the z-axis are somewhat "strictly monotonic". A suitable metaphor illustrating this feature is a *honeycomb*. The partition volumes correspond to the cells, and borders correspond to the cell walls.

3.2 Visualization of Spatio-Temporal Partitions

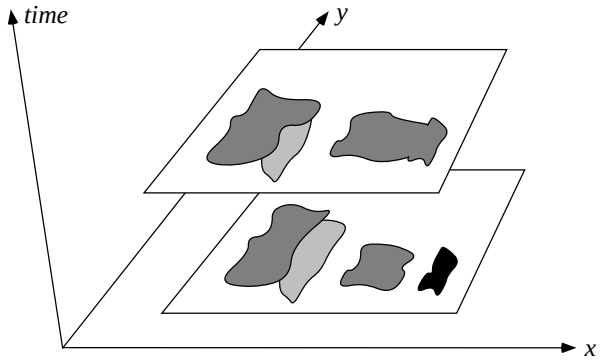
The three-dimensional object view suggests an obvious method to visualize spatio-temporal partitions, namely by three-dimensional pictures. In particular, it views spatio-temporal partitions from a global perspective since one can completely see the whole evolution of a spatial partition. Nevertheless, it seems that 3D pictures are too static and that they are only adequate for simple spatio-temporal partitions with a small number of regions and little changes.

An alternative which can be especially used to grasp complex application scenarios like weather reports are methods of virtual reality like cyber gloves

which allow to migrate through space and time. Another technique is to record snapshots in temporal order like on a film and to play the film. This is already used, for example, in weather forecasting where the development of temperature zones is displayed in a film. The disadvantage is that the user has no control of how the film is played.

An improvement of the snapshot approach is to admit user interaction when the film is displayed. A user interface could, for example, contain a time slider which allows to exert influence on the time interval, on the speed, and on the temporal direction of display and which gives an immediate feedback with respect to space and time.

A more sophisticated strategy could take a layered approach. The idea is to visualize two or more snapshots of the same scenario at different time points in parallel. For that purpose, the representations of these snapshots have to be slightly bent in the same way and to be positioned one on top of the other in temporal order. The spatial distance between two subsequent snapshot layers should visualize their temporal distance. This strategy allows to simultaneously compare the development of a spatial partition at different times. Each snapshot can be either controlled individually by its own time slider or altogether with the others to maintain selected temporal distances between subsequent snapshots.



3.3 Operations on Spatio-Temporal Partitions

In this section we focus on the transfer of the two basic spatial partition operations *intersection* and *relabel* to the spatio-temporal case and additionally introduce some new operations that are more directed to the time dimension. All operations are discussed informally and illustrated with query examples. A spatio-temporal *refine* poses problems, as we will see later, and is thus omitted. Moreover, in [6] *refine* was mainly employed to define the *window* operation.

Overlay. Similar to the spatial case, temporal overlays are based on a spatio-temporal *intersection* operation and turn out to be the most important operations on spatio-temporal partitions. They can be used to analyze the temporal evolution of two (or more) different attribute categories.

For example, consider a temporal map indicating the extent of mineral resources like oil fields or coal deposits and another temporal map showing the country map over time. Then an overlay of both temporal maps can, for instance, reveal the countries that had or still have the richest mineral resources, it can show the grade of decline of mineral deposits in the different countries, and it can expose the countries which most exploited their mineral resources.

Another application refers to social analysis and assumes temporal maps about average income, about the countries of the world, and about ethnic groups. If we overlay these three temporal maps (by two intersections), we can, for instance, recognize which ethnic groups in which countries belong to the richest or poorest social strata and whether the same ethnic groups have the same social status in different countries.

A further example for overlay supports weather forecasting and gale warning. Suppose that we are given three temporal maps, one about temperature distributions, one about high/low pressure areas, and one about spatial and temporal occurrences of storms and hurricanes. An overlay of these three temporal maps gives information about the influence of temperature and air pressure on the formation of hurricanes.

Clipping. An interesting special case of *intersection* is the *clipping* operation. We transfer the 2D operation to the spatio-temporal case but admit not only constant *spatio-temporal unit partitions* as "clipping volumes" but also general, that is, time-varying, ones. An application is a temporal map about the development of diseases. As a clipping window we use a temporal map of urban areas developing in space over time. The task is to analyze whether there is a connection between the increase or decrease of urban space and the development of certain diseases. Hence, all areas of disease outside of urban regions are excluded from consideration. The clipping works as a *spatio-temporal filter*. Another example is a temporal map about animal species that predominantly populate certain areas. As a clipping window we use a temporal map of forests and ask for the development of animal species in growing and shrinking forests.

Reclassification. As an application of a spatio-temporal *relabel* operation we take a temporal map marking all countries of the world with their population numbers. This allows to pose a ranking of countries with regard to their population number for each time and especially over time. A query can now ask for the proportion of each country's population on the world population over time, a task that can be performed by temporal relabeling. This corresponds to a reclassification of attribute categories over time without changing geometry.

Fusion. Assume that a temporal map of districts with their land use is given. The task is to identify regions with the same land use over time. At each time neighboring districts with the same land use are replaced by a single region, that is, their common boundary line is erased. We obtain a temporal fusion operator which is based on relabeling.

Static and Dynamic Relabeling. In the two previous examples the relabeling function did not change over time, that is, it was constant. We call this *static relabeling*. But we can generalize even this and look at applications requiring temporally changing relabeling functions. It can often be used if the semantics of attribute classifications alters over time. An example of *dynamic relabeling* is the classification of income to show poor and rich areas over time. Due to the changing value of money, due to inflation, and due to social changes, the understanding of wealthy and poorness varies over time. Hence, we need different and appropriate relabeling functions that are applied to distinct time intervals.

Dynamic relabeling can also be used for temperature maps. Imagine we have two relabeling functions, the first function mapping different temperature zones to distinct warm colors (orange, red, yellow, etc.) and a second function mapping temperatures less than five degrees to dark blue and the other temperatures to light blue. A visualization of temperature maps could now use the first relabeling (presentation) function for daytime and the second relabeling function for the night.

Domain. The two operations *intersection* and *relabel* are motivated by spatial aspects. In order to also emphasize temporal aspects, we add a few more operations addressing the time dimension. The operation *dom* determines the domain of a temporal map, that is, all times where the map does not yield the completely undefined partition. An example is a temporal map of earthquakes and volcanic eruptions in the world as they are interesting for seismological investigations. Applying the operation *dom* on this map returns the time periods of earthquake and volcanic activity in the world.

Temporal restriction. The operation *restrict* realizes a function restriction on spatio-temporal partitions and computes a new partition. As parameters it obtains a temporal map and a set of (right half-open) time intervals describing the time periods of interest. Imagine that we have a temporal map of birth rates, and we are only interested in the birth rates between 1989 and 1991 and between 1999 and 2001 ("millennium baby"). Then we can exclude all the other time periods and compare the change of birth rates in these two time intervals.

Temporal selection. The operation *select* is also a very powerful operation. It allows to scan spatio-temporal partitions over time and to check for each time whether a specified predicate is fulfilled or not. Consider a map showing the spread of fires. We could be interested in when and where the spread of fires occupied an area larger than 300 km^2. The operation *select* takes the temporal map and an appropriate predicate as arguments and computes the resulting temporal map (whose domain is, in general, reduced). Other applications are when and in which districts the birth rates exceeded the death rates, or when and in which countries socialist/conservative parties ruled.

Temporal aggregation. The operation *aggregate* collects all labels of a point over time and combines them with the aid of a binary function into one label. The result is a two-dimensional spatial partition. If, for example, the population numbers, the birth rates, the death rates, the population density, the average income, etc. of the countries in the world are available, we can aggregate over them and compute the maximum or minimum value each country ever had for one of these attributes.

Temporal projection. A special kind of aggregation is realized by the *project* operation which computes the projection of a spatio-temporal partition onto the Euclidean space and which yields a spatial partition. For each point in space, all labels, except for the undefined label denoting the outside, are collected over time. That is, if a point has always had the same single label over its lifetime, this single label will appear in the resulting partition and indicate a place that has never changed. On the other hand, points of the resulting partition with a collection of labels describe places where changes occurred.

Another application is the projection of a temporal map showing which countries were ruled by which parties. The result reveals those countries that were always ruled by the same party and those countries that had to accept changes of ruling parties. A further example is the projection of a temporal map illustrating the water levels of lakes onto the Euclidean plane. The result shows those parts of lakes that have always, sometimes, and never been covered with water.

4 A Formal Model of Spatial Partitions

In this section we briefly repeat some definitions of our model for two-dimensional spatial partitions. First, we provide a precise definition for the type of two-dimensional partitions in Section 4.1 since the snapshot of a spatio-temporal partition at a certain point in time (that is, the application of a spatio-temporal partition to a time point) yields a two-dimensional partition. After that we define the operations on two-dimensional partitions in Section 4.2. These operations are used in Section 5.2 to define operations on spatio-temporal partitions. Three-dimensional spatial partitions can be defined as a generalization of two-dimensional ones; they provide a helpful model to understand spatio-temporal partitions, and they can actually serve as a specification for an implementation of spatio-temporal partitions. Exhibiting the precise relationships between spatio-temporal partitions and 3D partitions will be a topic of future research.

Before we start giving mathematical definitions for partitions, we shortly summarize the employed notation. The application of a function $f : A \to B$ to a set of values $S \subseteq A$ is defined as $f(S) := \{f(x) \mid x \in S\} \subseteq B$. If we are sure that $f(S)$ yields a singleton set, we write $f^!(S)$ to denote this single element (instead of the singleton set), that is, $f(S) = \{y\} \implies f^!(S) = y$. ($f^!(S)$ is undefined if $|f(S)| \neq 1$.) Similarly, for doubly-nested singleton sets we use $f^{!!}$ to extract elements, that is, $f(S) = \{\{y\}\} \implies f^{!!}(S) = y$. The *inverse function* $f^{-1} : B \to 2^A$ of f is defined by $f^{-1}(y) := \{x \in S \mid f(x) = y\}$. Note that,

in general, f^{-1} is a partial function and that f^{-1} applied to a set yields a set of sets. The *range* of a function $f : A \to B$ is defined as $rng(f) := f(A)$. We also introduce a notation for power sets containing sets of constrained size: for $\triangleright \in \{>, \geq, =\}$ and $k \in \mathbb{N}$ we define $S^{\triangleright k} := \{s \in S \mid |s| \triangleright k\}$.

Let (X, T) be a topological space with topology $T \subseteq 2^X$, and let $S \subseteq X$.[1] The *interior* of S is defined as the union of all open sets that are contained in S and is denoted by $\operatorname{Int} S$, and the *closure* of S is defined as the intersection of all closed sets that contain S and is denoted by $\overline{S}$. The *exterior* of S is given by $\operatorname{Ext} S := \operatorname{Int}(X - S)$, and the *boundary* (or *frontier*) of S is defined as $\operatorname{Fr} S := \overline{S} \cap \overline{X - S}$. An open set is called *regular* if $A = \operatorname{Int} \overline{A}$. Regular open sets are closed under intersection. A topological space we need in this paper is $\mathbb{R}^2$.

A *partition* of a set S can be viewed as a total function $f : S \to I$ into an index set I. f induces an equivalence relationship $\equiv_f$ on S that is defined by $x \equiv_f y \iff f(x) = f(y)$. The equivalence classes $S/\equiv_f$ are called *blocks*. The block S_i that corresponds to an index i is given by $S_i := f^{-1}(i)$, and the whole partition $\{S_i \mid i \in I\}$ $(= S/\equiv_f)$ is also given by $f^{-1}(I)$ if f is surjective.

4.1 Two-Dimensional Spatial Partitions

A spatial partition is not just defined as a function $f : \mathbb{R}^2 \to I$ for two reasons: first, in most applications f cannot be assumed to be total, and second, f cannot be uniquely defined on borders between adjacent subsets of $\mathbb{R}^2$. Moreover, it is desirable from an application point of view to require blocks (modeling regions of a common label) to be regular open sets [19].

Therefore, we have defined spatial partitions in several steps [6]: first, a *spatial mapping* of type A is a total function $\pi : \mathbb{R}^2 \to 2^A$. We require the existence of an undefined element $\perp_A \in A$, which is used to represent undefined labels, that is, the "exterior" or "outside" of a partition is the block $b \subseteq \mathbb{R}^2$ with $\pi^!(p) = \perp_A$ for all $p \in b$. The power set range type is used to model labels on region borders: a *region* of π is a block that is mapped to a singleton set whereas a *border* of π is a block that is mapped to a subset of A containing two or more elements. Then the *interior* of π is defined as the union of π's regions, and the *boundary* of π is defined as the union of π's borders.

Definition 1. Let π be a spatial mapping of type A.

(i) $\rho(\pi) := \pi^{-1}(rng(\pi)^{=1})$	(*regions*)
(ii) $\omega(\pi) := \pi^{-1}(rng(\pi)^{>1})$	(*borders*)
(iii) $\iota(\pi) := \bigcup_{r \in \rho(\pi)} r$	(*interior*)
(iv) $\beta(\pi) := \bigcup_{b \in \omega(\pi)} b$	(*boundary*)

Finally, a *spatial partition* of type A is a spatial mapping of type A whose regions are regular open sets and whose borders are labeled with the union of labels of all adjacent regions:

[1] Recall that in a topological space the following three axioms hold [3]: (i) $U, V \in T \implies U \cap V \in T$, (ii) $S \subseteq T \implies \bigcup_{U \in S} U \in T$, and (iii) $x \in T$, $\varnothing \in T$. The elements of T are called *open sets*, their complements in X are called *closed sets*, and the elements of X are called *points*.

Definition 2. A *spatial partition* of type A is a spatial mapping π of type A with:
(i) $\forall r \in \rho(\pi) : r = \mathrm{Int}\,\overline{r}$
(ii) $\forall b \in \omega(\pi) : \pi^{!}(b) = \{\pi^{!!}(r) \mid r \in \rho(\pi) \wedge b \subseteq \overline{r}\}$
The type of all partitions of type A is denoted by $[A]$.

The reader might wonder why in the above definition (and in some of the following) we have used $\pi^{!}(b)$ on the left hand side and $\pi^{!!}(r)$ on the right hand side in (ii). This is explained in the following remark to which we will refer quite a few times later:

Remark 1

Consider a block of a partition π, for example, a region r or a border b. For each point p that is contained in r or in b, $\pi(p)$ yields as a label a set of values. For $p \in r$, this is a singleton set, say $\{a_1\}$, and for $p \in b$, this is a set $\{a_1, a_2, \dots\}$ of two or more elements. Now when we apply π to the whole set r (or b), we obtain the set of all labels for all points. By definition these are all equal, so the results of $\pi(r)$ and $\pi(b)$ are $\{\{a_1\}\}$ and $\{\{a_1, a_2, \dots\}\}$, respectively. Thus, if we want to denote the common label of all points of a block, this is given by $\pi^{!}(r) = \{a_1\}$ or $\pi^{!}(b) = \{a_1, a_2, \dots\}$, respectively. Likewise, $\pi^{!!}(r) = a_1$.

Hence, $\pi^{!}(b)$ denotes the common label, a set $\{a_1, a_2, \dots\}$, of border block b, and $\pi^{!!}(r)$ gives the label a_i of each touching region.

In the following we will sometimes need the notion of a *constant partition* which is a partition that yields one and the same label for all points:

$$\pi_x = \lambda p{:}\mathbb{R}^2.x$$

As a special case, the undefined partition of type A is given by $\pi_{\perp_A}$. Note that the lambda-notation $\lambda x{:}S.e(x)$ is just a shorthand for the set expression $\{(x, e(x)) \mid x \in S\}$ (which actually represents a function).

4.2 Operations on 2D-Partitions

We have defined three basic operations on spatial partitions: *intersection*, *relabel*, and *refine*. The intersection of two partitions π_1 and π_2 of types A and B, respectively, is again a spatial partition (of type $A \times B$) where each interior point p is mapped to the pair of values $(\pi_1^{!}(p), \pi_2^{!}(p))$, and all border points are mapped to the set of labels of all adjacent regions (as required by the second part of the definition of partition). Formally, we can define the intersection of two partitions $\pi_1 : [A]$ and $\pi_2 : [B]$ in several steps: first, we compute the regions of the resulting partition. This can be done by simple set intersection since regions are, by definition, regular open sets and since $\cap$ is closed on regular open sets:

$$\rho_{\cap}(\pi_1, \pi_2) := \{r \cap r' \mid r \in \rho(\pi_1) \wedge r' \in \rho(\pi_2)\}$$

Second, the union of all these regions gives the interior of the resulting partition:

$$\iota_\cap(\pi_1, \pi_2) := \bigcup_{r \in \rho_\cap(\pi_1,\pi_2)} r$$

Now the spatial mapping restricted to the interior can be just obtained by mapping each interior point $p \in I := \iota_\cap(\pi_1, \pi_2)$ to the pair of labels given by π_1 and π_2, respectively:

$$\pi_I := \lambda p{:}I.\{(\pi_1^!(p), \pi_2^!(p))\}$$

Third, the boundary labels can be derived from the labels of all adjacent regions. Let $R := \rho_\cap(\pi_1, \pi_2)$, $I := \iota_\cap(\pi_1, \pi_2)$, and $F := \mathbb{R}^2 - I$. Then we have:

$$\begin{aligned} &\mathit{intersection} : [A] \times [B] \to [A \times B] \\ &\mathit{intersection}(\pi_1, \pi_2) := \pi_I \cup \lambda p{:}F.\{\pi_I^{!!}(r) \mid r \in R \wedge p \in \overline{r}\} \end{aligned}$$

To understand the use of $\pi^{!!}$ in the above definition, recall Remark 1: since we have to place pairs of labels in the result set and since $\pi(r) = \{\{(a_1, a_2)\}\}$, we obtain (a_1, a_2) by application of $\pi^{!!}$.

The fact that *intersection* indeed yields a spatial partition is captured by the following lemma:

Lemma 1. *If* $\pi_1 : [A]$ *and* $\pi_2 : [B]$*, then* $\mathit{intersection}(\pi_1, \pi_2) : [A \times B]$. □

The proof can be found in [6].

Relabeling a partition π of type A by a function $f : A \to B$ is defined as $f \circ \pi$, that is, in the resulting partition of type B each point p, interior as well as boundary, is mapped to $f(\pi(p))$:

$$\begin{aligned} &\mathit{relabel} : [A] \times (A \to B) \to [B] \\ &\mathit{relabel}(\pi, f) := \lambda p{:}\mathbb{R}^2.f(\pi(p)) \end{aligned}$$

The fact that *relabel* is well-defined and always yields a spatial partition can be proved in several steps culminating in

Lemma 2. *If* $\pi : [A]$ *and* $f : A \to B$*, then* $\mathit{relabel}(\pi, f) : [B]$. □

Again, the proof is given in [6].

Finally, the refinement of a partition means the identification of connected components. This is achieved by attaching consecutive numbers to the components. We omit the formal definition here (see [6] for details) since we are not going to define spatio-temporal refinement anyhow.

5 Spatio-Temporal Partitions

In the following we have to work with different kinds of partitions. We therefore add the subscripts 2D and ST to disambiguate notations. For example, π_{2D} denotes a two-dimensional partition, and $[A]_{ST}$ denotes the set of all spatio-temporal partitions of type A. When no subscript is given, we assume by default spatio-temporal partitions, that is, π is a shorthand for π_{ST}.

5.1 The Type of Spatio-Temporal Partitions

In order to define temporally changing partitions we need a type for time. Since we are dealing with continuously changing information, we use $time = \mathbb{R}$ as a model of time.

Next we define a spatio-temporal partition as a function of two-dimensional spatial partitions over time. A *spatio-temporal mapping* of type A is a total function $\pi_{ST} : time \to [A]_{2D}$. Spatio-temporal mappings are too general to be used as a partition model since they can have an undesired structure: consider, for example, two different two-dimensional partitions π_{2D} and π'_{2D} and the following spatio-temporal mapping:

$$\pi_{ST}(t) \;=\; \begin{cases} \pi_{2D} & \text{if } t \text{ is rational} \\ \pi'_{2D} & \text{otherwise} \end{cases}$$

Here π_{ST} describes a completely discontinuous, rather pathological, change of partitions; we would like to rule out such spatio-temporal mappings as spatio-temporal partitions.

What we require is a kind of (semi-)continuity of spatio-temporal mappings. More precisely, we regard only *upper semicontinuous* spatial mappings as spatio-temporal partitions. To formally define continuity we employ a difference measure for 2D spatial partitions. A measure of "nearness" or "equality" is given by the size of the total area that is labeled equally in both partitions (that is, the corresponding set of points). We can now define an operator $\delta : [A]_{2D} \times [A]_{2D} \to \mathbb{R}$ that computes the size of the regions that are labeled differently.

$$\delta(\pi, \pi') := \int\limits_{\{p \in \mathbb{R}^2 \mid \pi(p) \neq \pi'(p)\}} dx\, dy$$

Using δ we can define the notion of upper semicontinuity of a spatial mapping.

Definition 3. Let π be a spatio-temporal mapping and $t \in time$. π is *upper semicontinuous* at t if $\lim_{\epsilon \to 0} \delta(\pi(t), \pi(t+\epsilon)) = 0$. Moreover, π is *upper semicontinuous* (everywhere) if it is upper semicontinuous at each $t \in time$.

Now we accept as spatio-temporal partitions only upper semicontinuous spatio-temporal mappings.

Definition 4. A *spatio-temporal partition* of type A is an upper semicontinuous spatio-temporal mapping of type A.

The type of spatio-temporal partitions with labels of type A is denoted by $[A]_{ST}$.

5.2 Basic Operations on Spatio-Temporal Partitions

Next we define the two partition operations *intersection* and *relabel*. We omit a definition of spatio-temporal *refine*, since it leads to a somewhat unpleasant behavior of the numbering of blocks. For example, a spatial partition $\pi_{2D} = \pi_{ST}(t)$

obtained by a refined spatio-temporal partition π_{ST} contains, in general, non-consecutive numberings of labels. Moreover, we need *refine* mainly for completeness reasons (that is, to be able to define the application-specific operation "window", see [6]). Instead, we are considering some operations that take the time dimension more explicitly into account, namely, operations for temporal selection and for temporal projection/aggregation.

Since spatio-temporal partitions are defined as temporal functions of 2D partitions, we can simply reduce the definitions to the two-dimensional case. For example, for intersection we obtain:

$$\begin{aligned} &\mathit{intersection}_{ST} : [A]_{ST} \times [B]_{ST} \to [A \times B]_{ST} \\ &\mathit{intersection}_{ST}(\pi, \pi') := \lambda t{:}\mathit{time}.\mathit{intersection}_{2D}(\pi(t), \pi'(t)) \end{aligned}$$

We have to prove that this definition indeed yields a spatio-temporal partition as a result.

Lemma 3. *If* $\pi : [A]_{ST}$ *and* $\pi' : [B]_{ST}$, *then* $\mathit{intersection}_{ST}(\pi, \pi') : [A \times B]_{ST}$.

Proof. Let $\pi_I = \mathit{intersection}_{ST}(\pi, \pi')$. First, it is obvious from the definition that π_I is a spatio-temporal mapping (of type $A \times B$). It remains to be shown that π_I is upper semicontinuous, which means to show that $\lim_{\epsilon \to 0} \delta(\pi_I(t), \pi_I(t+\epsilon)) = 0$. We call the points that are labeled differently by $\pi_I(t)$ and $\pi_I(t+\epsilon)$ the *difference region* for π_I (on the interval $[t, t+\epsilon]$). We next show that a difference region for π_I is always covered by the difference regions for π and for π'.

Consider an arbitrary point $p \in \mathbb{R}^2$, a time value $t \in \mathit{time}$, and an $\epsilon > 0$. By definition of δ, p is in the difference region for π_I if $\pi_I(t)(p) \neq \pi_I(t+\epsilon)(p)$. By the definition of $\mathit{intersection}_{ST}$ we know $\pi_I(t) = \mathit{intersection}_{2D}(\pi(t), \pi'(t))$, and thus we have:

$$\begin{aligned} &\pi_I(t)(p) \neq \pi_I(t+\epsilon)(p) \iff \\ &\quad \mathit{intersection}_{2D}(\pi(t), \pi'(t))(p) \neq \mathit{intersection}_{2D}(\pi(t+\epsilon), \pi'(t+\epsilon))(p) \end{aligned}$$

Now the definition of $\mathit{intersection}_{2D}$ depends on whether p is an interior or a boundary point. We can ignore boundary points, since they do not contribute to the value of the integral in δ. (This is because the area (2D) integral of lines (1D) is always 0.) Since p can be an interior point of $\pi_I(t)$ only if it is an interior point of both $\pi(t)$ and $\pi'(t)$, we need to consider the definition of $\mathit{intersection}_{2D}$ only for interior points. Therefore, we can substitute the definition for interior points into the above condition:

$$\pi_I(t)(p) \neq \pi_I(t+\epsilon)(p) \iff (\pi(t)(p), \pi'(t)(p)) \neq (\pi(t+\epsilon)(p), \pi'(t+\epsilon)(p))$$

Next we observe that the inequality on the right hand side holds if either of the pairs' components are not equal, that is,

$$\pi_I(t)(p) \neq \pi_I(t+\epsilon)(p) \iff \pi(t)(p) \neq \pi(t+\epsilon)(p) \vee \pi'(t)(p) \neq \pi'(t+\epsilon)(p)$$

Now this is nothing but the condition that p is either contained in the difference region for π or for π'.

We can conclude that if the difference region for π_I is covered by the difference regions for π and for π', the value of $\lim_{\epsilon\to 0}\delta(\pi_I(t),\pi_I(t+\epsilon))$ is bounded by

$$\lim_{\epsilon\to 0}\delta(\pi(t),\pi(t+\epsilon)) + \lim_{\epsilon\to 0}\delta(\pi'(t),\pi'(t+\epsilon))$$

Since both terms are 0, we also know that $\lim_{\epsilon\to 0}\delta(\pi_I(t),\pi_I(t+\epsilon)) = 0$. □

Next we define the relabeling operation. A first attempt is to simply "lift" the two-dimensional function as follows.

$$\begin{aligned}&\mathit{relabel}_{ST} : [A]_{ST} \times (A \to B) \to [B]_{ST}\\ &\mathit{relabel}_{ST}(\pi, f) := \lambda t{:}\mathit{time}.\mathit{relabel}_{2D}(\pi(t), f))\end{aligned}$$

Note that the relabeling function f must be total. We can generalize this definition considerably by allowing a temporally changing relabeling function. Thus, we could try the slightly changed definition shown below:[2]

$$\begin{aligned}&\mathit{relabel}_{ST} : [A]_{ST} \times (\mathit{time} \to A \to B) \to [B]_{ST}\\ &\mathit{relabel}_{ST}(\pi, f) := \lambda t{:}\mathit{time}.\mathit{relabel}_{2D}(\pi(t), f(t)))\end{aligned}$$

Unfortunately, with this definition, we can construct spatio-temporal mappings that are not spatio-temporal partitions. As a simple example consider the "universal partition" $\underline{\pi_a}$ $(= \lambda t{:}\mathit{time}.\lambda p{:}\mathbb{R}^2.a)$ that maps every point at every time to $a \in A := \{a, b, \bot_A\}$. We relabel $\underline{\pi_a}$ with the function $f : \mathit{time} \to A \to A$ that is defined by:

$$f(t)(x) \;=\; \begin{cases} b & \text{if } t = t_0\\ a & \text{otherwise}\end{cases}$$

so that $\mathit{relabel}(\pi_a, f)$ yields a spatio-temporal mapping π° which is defined by:

$$\pi^\circ(t)(p) \;=\; \begin{cases} b & \text{if } t = t_0\\ a & \text{otherwise}\end{cases}$$

Now it is obvious that π° is not upper semicontinuous at t_0 and is thus not a spatio-temporal partition. Therefore, the above definition for *relabel* is too general.

Fortunately, by adding a simple condition we can ensure that *relabel* again yields spatio-temporal partitions: we require f to be what we call *upper semi-constant*, that is, we demand $\forall t \in \mathit{time} : \exists\epsilon > 0 : \forall 0 \le \delta \le \epsilon : f(t+\delta) = f(t)$. This means that after each change, f must be constant for some short period of time. (This rules out dynamic changes of relabeling function that would, in principle, also be possible. However, to define the more general version we need a quite complex topological continuity definition for function spaces requiring A and B to be topological spaces, too, which is not needed otherwise. In summary, it seems that there are not very many highly important applications for the general case so that the complex definition does not seem to be justified.)

[2] Note that the function type constructor associates to the right, that is, $X \to Y \to Z = X \to (Y \to Z)$.

5.3 More Operations on Spatio-Temporal Partitions

Beyond the "classical" partition operations there are some more operations that address the time dimension more explicitly. First of all, we can determine the domain of a spatio-temporal partition π which is defined as the set of all times t when $\pi(t)$ does not yield the undefined 2D partition.

$$\begin{aligned} &dom_{ST} : [A]_{ST} \to 2^{time} \\ &dom_{ST}(\pi) := \{t \in time \mid \pi(t) \neq \pi_{\perp_A}\} \end{aligned}$$

Related to *dom* is the operation *restrict* that restricts the domain of a spatio-temporal partition to a subset of *time*. For the same reasons as for *relabel* we cannot allow arbitrary subsets of *time*. Instead we require the set be given by *right half-open* intervals. For a totally ordered set S, right half-open intervals are defined as:

$$[s, t[:= \{x \in S \mid s \leq x < t\}$$

The set of all these intervals is then defined by $[S[:= \{[s, t[\mid s, t \in S\}$. In practice we need sets of intervals that do not overlap (to describe subsets of *time*):

$$\langle S \rangle := \{T \in 2^{[S[} \mid \forall I, J \in T : I \cap J \neq \varnothing \implies I = J\}$$

The set of values contained in an interval set S is denoted by $\cup S := \bigcup_{s \in S} s$. Now we can define the *restrict* operation precisely.

$$\begin{aligned} &restrict : [A]_{ST} \times \langle time \rangle \to [A]_{ST} \\ &restrict(\pi, T) := \lambda t{:}time. \begin{cases} \pi(t) & \text{if } t \in \cup T \\ \pi_{\perp_A} & \text{otherwise} \end{cases} \end{aligned}$$

To prove that *restrict* indeed yields spatio-temporal partitions as results we exploit the fact that *restrict* can also be defined via *relabel*: let "id" denote the identity function, that is, $\mathrm{id}(x) = x$. We have:

Lemma 4.
Let $T : \langle time \rangle$ *and* $f_T := \begin{cases} \mathrm{id} & \text{if } t \in \cup T \\ \lambda x{:}A.\perp_A & \text{otherwise} \end{cases}$. *Then:*
$restrict(\pi, T) = relabel(\pi, f_T)$ □

Since f_T is upper semiconstant by its definition, we obtain as a corollary of Lemma 4:

Corollary 1. *If* $\pi : [A]_{ST}$ *and* $T : \langle time \rangle$, *then* $restrict(\pi, T) : [A]_{ST}$. □

The notion of restriction of spatio-temporal partitions can be also viewed from a different perspective: we can restrict a partition to those times at which a predicate on the corresponding spatial partition is true. This is a kind of "temporal selection" which could be formally defined by:

$$
\begin{aligned}
&\mathit{select} : [A]_{ST} \times ([A]_{2D} \to \mathbb{B}) \to [A]_{ST} \\
&\mathit{select}(\pi, P) := \lambda t{:}\mathit{time}. \begin{cases} \pi(t) & \text{if } P(\pi(t)) \\ \pi_{\perp_A} & \text{otherwise} \end{cases}
\end{aligned}
$$

Unfortunately, without any restriction on the kind of partition predicates used as arguments this definition does not, in general, yield spatio-temporal partitions as a result. As an example consider a spatio-temporal partition π that describes a constantly shrinking square labeled by a (this can be well imagined as a pyramid) and a predicate P that asks for the size of the region labeled a to be exactly x. Assuming that the square is initially greater and at the end smaller than x, an expression $\mathit{select}(\pi, P)$ yields a spatial mapping (much like the one described for *relabel*) that gives only at one time t a spatial partition other than $\pi_{\perp_A}$ (namely a square of size x). Thus, $\mathit{select}(\pi, P)$ is not upper semicontinuous at t and is therefore not a spatio-temporal partition.

How can we correct the definition? It seems to be extremely difficult to characterize the class of predicates that guarantee the above definition to deliver proper spatio-temporal partitions. Fortunately, we can take a different route: first, we determine the set of time intervals on which P is true. This is done by finding the set of time points at which P holds and then regularizing this set by restricting it to half-open intervals. The time set regularization is performed by the function *reg* (note that $\forall I, J \in \langle \mathit{time} \rangle : I \geq J \iff \cup I \supseteq \cup J$):

$$
\begin{aligned}
&\mathit{reg} : 2^{\mathit{time}} \to \langle \mathit{time} \rangle \\
&\mathit{reg}(T) := \max\{I \in \langle \mathit{time} \rangle \mid \cup I \subseteq T\}
\end{aligned}
$$

Then we can define *select* simply via *restrict* on this regular interval set.

$$
\begin{aligned}
&\mathit{select} : [A]_{ST} \times ([A]_{2D} \to \mathbb{B}) \to [A]_{ST} \\
&\mathit{select}(\pi, P) := \mathit{restrict}(\pi, \mathit{reg}(\{t \in \mathit{time} \mid P(\pi(t))\}))
\end{aligned}
$$

With this definition we again get the proof that *select* computes spatio-temporal partitions for free.

Corollary 2. *If* $\pi : [A]_{ST}$ *and* $P : [A]_{2D} \to \mathbb{B}$, *then* $\mathit{select}(\pi, P) : [A]_{ST}$. □

Finally, we consider how to form aggregations of partitions over time. From a different point of view, a partition $\pi : [A]_{ST}$ can be also regarded as a function $\xi : \mathbb{R}^2 \to \mathit{time} \to A$, that is, ξ gives for each point $p \in \mathbb{R}^2$ a time-dependent label function ξ_p.[3] Aggregation now means to combine all the values delivered by ξ_p into a single label of type, say B, so that aggregation of a spatio-temporal partition yields a two-dimensional partition of type B.

We shall not give here a completely generic definition since infinite aggregations (over an infinite set of time points) lead to a quite complex definition because of several requirements on the label type. Therefore, we instead restrict to aggregating functions f of type $A \times A \to A$ that are commutative. Commutativity allows to process the labels in any order. Therefore, we can simply

[3] Formally, $\xi = \lambda p{:}\mathbb{R}^2.\lambda t{:}\mathit{time}.\pi(t)(p)$.

collect the set of all labels at a point p and aggregate them by f. For this we use the notation $agg(f, S)$ to denote the aggregation of a finite, non-empty set by a binary function:

$$\begin{aligned} agg(f, \{a\}) &= a \\ agg(f, (\{a\} \cup S)) &= f(a, agg(f, S)) \end{aligned}$$

Now we can define aggregation as follows:

$$\begin{aligned} &aggregate : [A]_{ST} \times (A \times A \to A) \to [A]_{2D} \\ &aggregate(\pi, f) := \lambda p{:}\mathbb{R}^2.agg(f, \cup_{s \in \{\pi(t)(p) \mid t \in time\}} s) \end{aligned}$$

Functions that can be used to get interesting aggregations are, for example, max or min. As a special case of aggregation (that actually omits aggregation altogether) we define the function *project* that simply collects all (defined) values for a point:

$$\begin{aligned} &project : [A]_{ST} \to [2^A]_{2D} \\ &project(\pi) := \lambda p{:}\mathbb{R}^2.\{\pi(t)(p) \mid t \in time\} - \{\bot_A\} \end{aligned}$$

This function exhibits very nicely a correspondence between spatio-temporal partitions and vagueness: assume we consider the growing/shrinking of lakes, forests, etc. over a certain period of time given by a corresponding partition π_{ST}. By computing $project(\pi_{ST})$ we obtain a two-dimensional spatial partition in which regions labeled with singleton sets like $\{\{a\}\}$ give regions that have not changed during the considered time interval, whereas labels $\{\{a\}, \{b\}, \{a, b\}\}$ indicate regions that do have changed (from $\{a\}$ to $\{b\}$). In the terminology of [7] the former regions correspond to the *kernel* and the latter to the *boundary* of a vague region.

6 Conclusions

We have investigated dynamically changing maps; in particular, we have identified and generalized operations that are of practical interest. Moreover, by formally defining spatio-temporal partitions as a generalization of spatial partitions we have provided a theoretical foundation for temporally changing maps and their operations. Thus, spatio-temporal partitions can serve as a formal backbone to dynamic maps as does the model of spatial partitions for static maps.

References

1. J. K. Berry. Fundamental Operations in Computer-Assisted Map Analysis. *Int. Journal of Geographical Information Systems*, 1(2):119–136, 1987.
2. T. S. Cheng and S. K. Gadia. A Pattern Matching Language for Spatio-Temporal Databases. In *ACM Conf. on Information and Knowledge Management*, pages 288–295, 1994.

3. J. Dugundji. *Topology.* Allyn and Bacon, 1966.
4. M. Erwig, R. H. Güting, M. Schneider, and M. Vazirgiannis. Abstract and Discrete Modeling of Spatio-Temporal Data Types. In *6th ACM Symp. on Geographic Information Systems*, pages 131–136, 1998.
5. M. Erwig, R. H. Güting, M. Schneider, and M. Vazirgiannis. Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. *GeoInformatica*, 3(3), 1999. To appear.
6. M. Erwig and M. Schneider. Partition and Conquer. In *3rd Int. Conf. on Spatial Information Theory*, LNCS 1329, pages 389–408, 1997.
7. M. Erwig and M. Schneider. Vague Regions. In *5th Int. Symp. on Advances in Spatial Databases*, LNCS 1262, pages 298–320, 1997.
8. M. Erwig, M. Schneider, and R. H. Güting. Temporal Objects for Spatio-Temporal Data Models and a Comparison of Their Representations. In *Int. Workshop on Advances in Database Technologies*, LNCS 1552, pages 454–465, 1998.
9. A. U. Frank. Overlay Processing in Spatial Information Systems. In *8th Int. Symp. on Computer-Assisted Cartography*, pages 16–31, 1987.
10. A. U. Frank, G. S. Volta, and M. MacGranaghan. Formalization of Families of Categorical Coverages. *Int. Journal of Geographical Information Science*, 11(3):215–231, 1997.
11. S. K. Gadia and S. S. Nair. Temporal Databases: A Prelude to Parametric Data. *In [18]*, pages 28–66, 1993.
12. R. H. Güting. Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems. In *Int. Conf. on Extending Database Technology*, LNCS 303, pages 506–527, 1988.
13. R. H. Güting and M. Schneider. Realm-Based Spatial Data Types: The ROSE Algebra. *VLDB Journal*, 4(2):100–143, 1995.
14. Z. Huang, P. Svensson, and H. Hauska. Solving Spatial Analysis Problems with GeoSAL, a Spatial Query Language. In *6th Int. Working Conf. on Scientific and Statistical Database Management*, 1992.
15. H.-P. Kriegel, T. Brinkhoff, and R. Schneider. The Combination of Spatial Access Methods and Computational Geometry in Geographic Database Systems. In *2nd Symp. on Advances in Spatial Databases*, LNCS 525, pages 5–21, 1991.
16. M. Schneider. *Spatial Data Types for Database Systems - Finite Resolution Geometry for Geographic Information Systems.* LNCS 1288. Springer-Verlag, 1997.
17. M. Scholl and A. Voisard. Thematic Map Modeling. In *1st Int. Symp. on Large Spatial Databases*, LNCS 409, pages 167–190, 1989.
18. A. U. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass. *Temporal Databases: Theory, Design, and Implementation.* The Benjamin/Cummings Publishing Company, 1993.
19. R. B. Tilove. Set Membership Classification: A Unified Approach to Geometric Intersection Problems. *The Computer Journal*, 37(1):25–34, 1994.
20. C. D. Tomlin. *Geographic Information Systems and Cartographic Modeling.* Prentice Hall, 1990.
21. G. S. Volta and M. J. Egenhofer. Interaction with Attribute Data Based on Categorical Coverages. In *1st Int. Conf. on Spatial Information Theory*, LNCS 716, pages 215–233, 1993.
22. M. F. Worboys. A Unified Model for Spatial and Temporal Information. *The Computer Journal*, 37(1):25–34, 1994.