In: *5th Int. Symp. on Advances in Spatial Databases (SSD'97)*, LNCS 1262, 298-320

Vague Regions

Martin Erwig & Markus Schneider

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

Abstract: In many geographical applications there is a need to model spatial phenomena not simply by sharp objects but rather through indeterminate or vague concepts. To support such applications we present a model of vague regions which covers and extends previous approaches. The formal framework is based on a general exact model of spatial data types. On the one hand, this simplifies the definition of the vague model since we can build upon already existing theory of spatial data types. On the other hand, this approach facilitates the migration from exact to vague models. Moreover, exact spatial data types are subsumed as a special case of the presented vague concepts. We present examples and show how they are represented within our framework. We give a formal definition of basic operations and predicates which particularly allow a more fine-grained investigation of spatial situations than in the pure exact case. We also demonstrate the integration of the presented concepts into an SQL-like query language.

1 Introduction

In the literature about spatial database systems and geographical information systems (GIS) that advocates an entity-oriented view of spatial phenomena, the general opinion prevails that special data types are necessary to model geometry and to efficiently represent geometric data in database systems, for example [Eg89, GNT91, GS93, GS95, Gü88, LN87, OM88, Sc95, SV89]. These data types are commonly denoted as *spatial data types* such as *point*, *line*, or *region*. We speak of *spatial objects* as occurrences of spatial data types.

So far, spatial data modeling implicitly assumes that the extent and hence the boundary of spatial objects is precisely determined and universally recognized. This leads exclusively to *exact object models*. Spatial objects are represented by sharply described points, lines, and regions in a defined reference frame. Lines link a series of exactly known coordinates (points), and regions are bounded by exactly defined lines which are called *boundaries*. The properties of the space at the points, along the lines, or within the regions are given by attributes whose values are assumed to be constant over the total extent of the objects. Examples are especially man-made spatial objects representing engineered artifacts (like highways, roads, houses, and bridges) and some predominantly immaterial spatial objects exerting social control (like countries and districts with their political and administrative boundaries or land parcels with their cadastral boundaries). We will denote this kind of entities as *determinate spatial objects*.

Increasingly, researchers are beginning to realize that there are many spatial objects in reality which do not have sharp boundaries or whose boundaries cannot be precisely determined. Examples are natural, social, or cultural phenomena like land features with continuously changing properties (such as population density, soil quality, vegetation), oceans, biotopes, deserts, an English speaking area, or mountains and valleys. The transition between a valley and a mountain usually cannot be exactly determined so that the two spatial objects "valley" and "mountain" cannot be precisely separated and defined. Frequently, the indeterminacy of spatial objects is associated with temporal changes; for example, clouds and sandbanks dynamically change their shapes in the course of time. We will denote this kind of entities as *vague* or *indeterminate spatial objects*.

This paper presents an object model for defining vague regions[1] which rests on "traditional" (that is, exact) modeling techniques. This modeling strategy simultaneously expresses the authors' opinion that it is unnecessary to begin from scratch when modeling vague spatial objects. On the contrary, it is possible to extend, rather than to replace, the current theory of spatial database systems and GIS. Furthermore, moving from an exact to a vague domain does not necessarily invalidate conventional geometry; it is merely an extension. Consequently, the current exact object models that are restricted to determinate spatial objects can be considered as simplified special cases of a richer class of models for general spatial objects. It turns out that this is exactly the case for the model to be presented.

Section 2 gives a characterization of the various meanings of indeterminacy, discusses the notion of "boundary", and presents a classification of the approaches proposed so far. Section 3 informally introduces the concept of vague regions and motivates the necessity of vague topological predicates and vague spatial operations. Section 4 formalizes these concepts and discusses the problem of adequately defining numerical operations on vague regions. Section 5 demonstrates an embedding into an SQL-like query language, and Section 6 draws some conclusions and gives a prospect of future research activities.

2 Classifying Models for Vague Spatial Objects

A first attempt of a taxonomy of vague spatial objects has been given by Couclelis [Co96]. She proposes to examine the essence of vague spatial objects from three different perspectives: the empirical nature of the object, the mode of observation, and the user's purpose. All three perspectives are based on the intuitive meaning of the notion "boundary". The nature of the object (for example, whether it is homogeneous or heterogeneous, continuous or discontinuous, solid or fluid, fixed or moving) influences how we become aware of the boundaries and their degree of sharpness. The mode of observation (given, for example, by scale, resolution, time, error) affects the knowledge about the position of the boundary. The user's purpose for which a model is designed leads to a preference for one model over the other, since different user categories have

1. Concepts for vague points and vague lines are currently not taken into account.

different requirements and conceptual views. Administrators, for instance, demand precisely defined objects; scientists, however, strive to integrate the vagueness of boundaries into their models.

The entity-oriented view of spatial phenomena, which we will take in this paper, considers spatial objects as conceptual and mathematical abstractions of real-world entities which can be identified and distinguished from the rest of space. For example, a region divides space into three parts: one part inside the object, another part on the border of the object, and the remaining part outside the object. The three parts form a partition of space, that is, they are mutually exclusive and covering the whole space. Hence, the notion of a region is intrinsically related to the notion of a boundary, be it sharp or indeterminate.

So far, in spatial data modeling boundaries are considered as sharp lines that represent abrupt changes of spatial phenomena and that describe and thereby distinguish regions with different characteristic features. The assumption of crisp boundaries harmonizes very well with the internal representation and processing of spatial objects in a computer which requires precise and unique internal structures. Hence, in the past, there has been a tendency to force reality into determinate objects. In practice, however, there is no apparent reason for the whole boundary of a region to be sharp or to have a constant degree of vagueness. There are a lot of geographical application examples illustrating that the boundaries of spatial objects can be indeterminate. For instance, boundaries of geological, soil, and vegetation units (see for example [Al94, Bu96, KV91, LAB96, WH96]) are often sharp in some places and vague in others; many human concepts like "the Indian Ocean" are implicitly vague.

The treatment of spatial objects with indeterminate boundaries is especially problematic for the computer scientist who is confronted with the difficulties how to model such objects in his database system so that they correspond to the user's intuition, how to finitely represent them in a computer format, how to develop spatial index structures for them, and how to draw them. He is accustomed to the abstraction process of simplifying spatial phenomena of the real world through the concepts of conventional binary logic, reduction of dimension, and cartographic generalization to precisely defined, simply structured, and sharply bounded objects of Euclidean geometry like points, lines, and regions.[2]

In reality, there are essentially two categories of indeterminate boundaries: sharp boundaries whose position and shape are unknown or cannot be measured precisely, and boundaries which are not well-defined or which are useless (for example, between a mountain and a valley) and where essentially the topological relationship between spatial objects is of interest.

2. Ironically, this abstraction process itself mapping reality onto a mathematical model implicitly introduces a certain kind of vagueness and imprecision.

Spatial objects with indeterminate boundaries are difficult to model and are so far not supported in spatial database systems. According to the two categories of boundaries, two kinds of vagueness or indeterminacy concerning spatial objects have to be distinguished: *Uncertainty* relates either to a lack of knowledge about the position and shape of an object with an existing, real border (*positional* uncertainty) or to the inability of measuring such an object precisely (*measurement* uncertainty). *Fuzziness* is an intrinsic feature of an object itself and describes the vagueness of an object which certainly has an extent but which inherently cannot or does not have a precisely definable border.

The subject of modeling spatial vagueness has so far been exclusively treated by geographers but rather neglected by computer scientists. At least three alternatives are proposed as general design methods:

- *fuzzy models* [Al94, Ba93, Bu96, Ed94, KV91, LAB96, Us96, Wa94, WHS90] which are all based on fuzzy set theory and predominantly model fuzziness,
- *probabilistic models* [Bl84, Bu96, Fi93, Sh93] which are based on probability theory and predominantly model positional and measurement uncertainty, and
- *exact models* [CF96, CG96, Sc96] which transfer data models, type systems, and concepts for spatial objects with sharp boundaries to spatial objects without clear boundaries and which predominantly model uncertainty but also aspects of fuzziness.

Fuzzy sets were first introduced by Zadeh [Za65] to treat imprecise concepts in a definable way. Fuzzy set theory is an extension or generalization (and not a replacement) of classical boolean set theory and deals only with fuzziness, not with uncertainty. Fuzziness is not a probabilistic attribute, in which the grade of membership of an individual in a set is connected to a given statistically defined probability function. Rather, it is an admission of the possibility that an individual is a member of a set or that a given statement is true. Examples of fuzzy spatial objects include mountains, valleys, biotopes, oceans, and many other geographic features which cannot be rigorously bounded by a sharp line.

Probability theory can be used to represent uncertainty. It defines the grade of membership of an entity in a set by a statistically defined probability function. Examples are the uncertainty about the spatial extent of particular entities like regions defined by some property such as temperature, or the water level of a lake.

The main difficulty of fuzzy and probabilistic models is that their use with spatial data is still a non-trivial application. On the one hand, our current computational technology does not allow efficient processing of uncertain and fuzzy spatial data. On the other hand, it is an open problem how to integrate and transform these models into the concept of spatial data types.

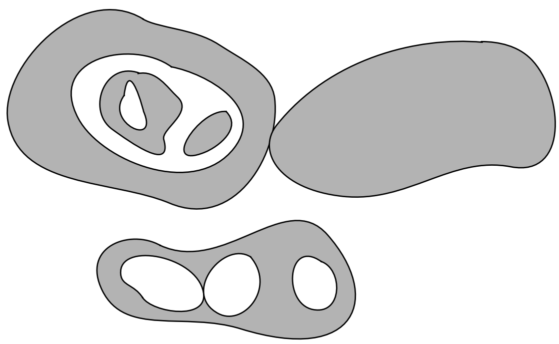
A benefit of the exact object model approach is that existing definitions, techniques, data structures, algorithms, etc., need not be redeveloped but only modified and extended, or simply used. The currently proposed exact methods model vague regions by using some kind of *zone* concept, either without holes [CF96, CG96] or with holes [Sc96]. The central idea is to consider determined zones surrounding the indeterminate boundaries of a region and expressing its minimal and maximal extension. The zones serve as a description and separation of the space that certainly belongs to the region and the space that is certainly outside.

While [CF96] and [CG96] are mainly interested in classifications of topological relationships between vague regions for which a simple model is assumed, [Sc96] proposes a model of complex vague regions with vague holes and focusses on their formal definition. Unfortunately, the three approaches are limited to "concentric" object models and have problems with geometric closure properties. The model described in this paper also pursues the exact model approach but is much more general and much simpler than the approaches suggested so far.

3 What are Vague Regions?

Our goal to base a concept of vague regions on traditional modeling techniques first necessitates a general exact object model for determinate regions. We will introduce this model only informally here. A formal definition of this model based on the point set paradigm and on point set topology is given in the Appendix. Each alternative model should fulfill the properties described there. Possible candidates are the models described in [ECF94, WB93], and the discrete model of the ROSE algebra [GS93, GS95, Sc95].

A (*determinate*) *region* is a set of disjoint, connected areal components possibly with disjoint holes (see the picture below). This model is very general and closed under (appropriately defined) geometric union, intersection, difference, and complement operations. It allows regions to contain holes and islands within holes to any (finite) level. The requirement of disjointedness is not meant in a strict sense; components of regions as well as holes of a component may be neighbored in a common boundary line or in common single boundary points.[3] We only require that the employed model satisfies the requirements defined in the Appendix.

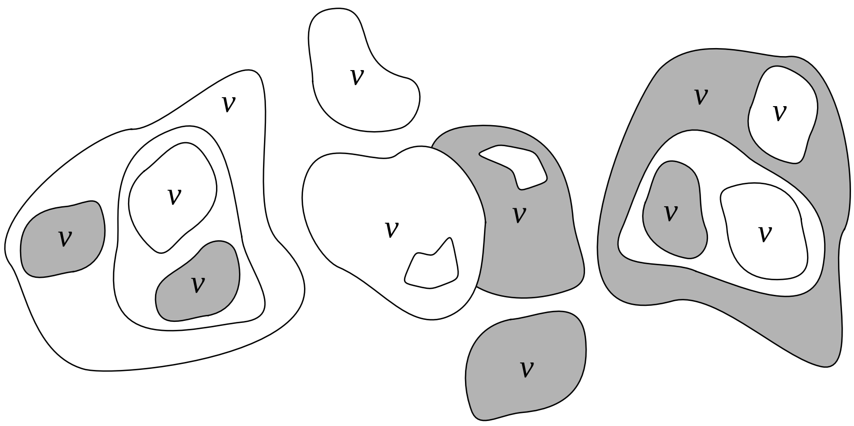


3. Usually, common boundary lines make no sense, since then adjacent components and adjacent holes, respectively, could be merged together by eliminating the common boundary parts. For our purposes, this aspect is not relevant.

Our concept of vague regions mainly deals with the aspect of uncertainty but also includes some aspects of fuzziness. Frequently, there is uncertainty about the spatial extent of phenomena in space, that is, objects can shrink and extend. An example is a lake whose water level depends on the amount of precipitation or on the degree of evaporation and which has thus a minimal and maximal extent. Another example is a map of natural resources like iron ore. For some areas experts definitely know the existence of iron ore because of soil samples. For other areas experts are not sure and only assume the incidence of this mineral. These are the kinds of vague regions we are especially interested in. On the other hand, our concept is also able to model the aspect of fuzziness that areal objects have an extent but cannot be bounded by a precise border, for example, the transition between a mountain and a valley. Continuous changes of features (like air pollution continuously decreasing from city centers to rural areas) cannot currently be modeled by this concept (but see Section 5).

A *vague region* is a pair of disjoint regions. The first region, called the *kernel,* describes the determinate part of the vague region, that is, the area which definitely and always belongs to the vague region. The second region, called the *boundary,* describes the vague part of the vague region, that is, the area for which we cannot say with any certainty whether it or parts of it belong to the vague region or not. *Maybe* the boundary or parts of it belong to the vague region, *maybe* this is not the case. Or we could say that this is *unknown*. It is important to notice that boundaries need not necessarily be one-dimensional structures but can be regions, and that the semantics of the boundary of a vague region is not fixed by our model but depends on the meaning the application associates with it.

The figure below gives an abstract example of a vague region *v*. The blank areas annotated with *v* depict kernels, the shaded areas annotated with *v* denote the boundaries of the vague region *v*, and the blank areas that are not annotated describe holes. The example demonstrates the complexity of the model. Kernels and boundaries may be adjacent; they may have holes which themselves can contain a hierarchy of kernels and boundaries with holes.

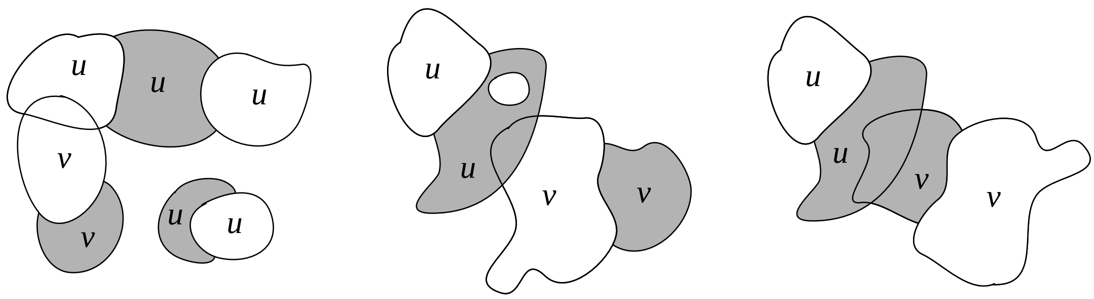


We now briefly present two real life applications and motivate the use of vague regions, vague topological predicates, and vague spatial operations. Vague concepts offer a greater flexibility for modeling properties of spatial phenomena in the real world than determinate concepts do. Still, vague concepts comprise the modeling power of determinate concepts as a special case.

The first example is taken from the animal kingdom and demonstrates the need of different *vague intersects* predicates and the use of a *vague intersection* operation. We view the living spaces of different animal species and distinguish kernel areas where they mainly live and boundary areas like peripheral areas or corridors where they in particular hunt for food or which they cross in order to migrate from one kernel area to another one. We now consider some relationships of their living spaces and ask:

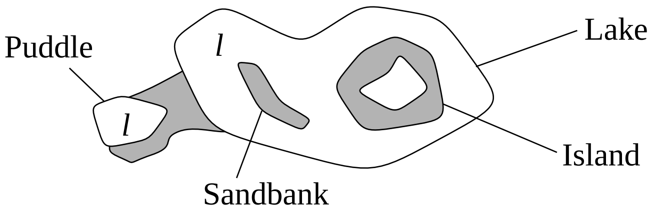
- Which animals (partially) share their living spaces?
- Which hunters penetrate into the living space of other animals?
- What are the areas where two species can only meet by accident?

For two animal species *u* and *v*, the interesting situations for the queries are shown below. They all relate to different kinds of intersection which amount to three different kinds of topological predicates (introduced in the next section). The first query asks for kernel/kernel intersections, the second query for kernel/boundary intersections but not kernel/kernel intersections, and the third query exclusively for boundary/boundary intersections. The situation on the left is definitely an intersection. In contrast, the situation in the middle is a vague intersection which, however, is a stronger case than the situation on the right. Other examples of topological relationships and their use will be presented in the next section.



The task to compute the common living spaces of two animal species asks for the intersection of two vague regions. The intersection of two kernels is certainly a kernel, and the intersection of an exterior part with anything else is an exterior part. The open question is now the intersection of a kernel with a boundary and the intersection of two boundaries. Since boundaries are vague, we cannot make a unique statement whether these intersections belong to the kernel parts or to the boundary parts. It only remains to regard these intersections as boundary parts.

The second example demonstrates that concentric models like those presented in [CF96, CG96, Sc96] are captured by our concept. Consider a lake *l* which has a minimal water level in dry periods (kernels) and a maximal water level in rainy periods. Dry periods can entail puddles. Small islands in the lake which are less flooded by water in dry and more (but never completely) flooded in rainy periods can be modeled through holes surrounded by a boundary. If an island like a sandbank can be flooded completely, it belongs to the boundary part.



4 An Exact Model of Vague Regions

In this section, we give a formal account of vague regions. We first define vague spatial operations in Section 4.1. After that we define predicates in Section 4.2. There we will see that a concept, such as *inside*, is not anymore simply a question of *true* and *false*, but rather needs a vague kind of booleans containing a value like *maybe*. That is, we actually employ a three-valued logic as the range of (standard) predicates. Similarly, numeric operations given in Section 4.3 seem to require a concept of vague numbers (given, for example, by intervals). Since this entails rather extensive changes to the type of real numbers and on its operations, we instead define different exact versions of numeric operations capturing various aspects of vagueness. In general, the problem is how to integrate vague regions with other types and operations of a data model. We will pick up this issue again in Section 6.

For the definition of vague regions we make use of a suitable model for determinate regions as sketched in the previous section. One possible candidate is the point set model the relevant parts of which are given in the Appendix. We can choose any other model as long as it offers the following operations (let R denote the type of regions and $\mathbb{R}$ the set of real numbers):

$\oplus : R \times R \to R$	(*union*)
$\otimes : R \times R \to R$	(*intersection*)
$\ominus : R \times R \to R$	(*difference*)
$\circleddash : R \to R$	(*complement*)
$dist : R \times R \to \mathbb{R}$	(*minimum distance*)
$area : R \to \mathbb{R}$	(*area*)

Moreover, R together with the operations $\oplus$ and $\otimes$ must form a boolean algebra. The order predicate of the corresponding boolean lattice is then given by $r \subseteq s \Leftrightarrow r \cup s = s$ ($\Leftrightarrow r \cap s = r$).

We define a *vague region* v as a pair of disjoint regions (k, b) where k gives the *kernel* of v and b denotes the *boundary* of v. We employ the following notation: $v^\kappa = k$ and $v^\beta = b$. Finally, the *exterior*, or *outside*, of v is defined as $v^\varepsilon = \circleddash(k \oplus b)$.

4.1 Vague Spatial Operations

In order to define operations, such as **union**, **intersection**, and **difference** of two vague regions *u* and *v*, it is helpful to consider the possible relationships between the kernel, boundary, and outside parts of *u* and *v*. We do this by giving a table for each operation where a column/row labeled by ●, ◍, or ○ denotes the kernel, boundary, or outside part of *u*/*v*. Each field of the table denotes a possible combination (that is, intersection) of kernel, boundary, and outside parts of both objects, and the label in each field specifies whether the corresponding intersection belongs to the kernel, boundary, or outside part of the operation's result.

union	●	◍	○
●	●	●	●
◍	●	◍	◍
○	●	◍	○

intersection	●	◍	○
●	●	◍	○
◍	◍	◍	○
○	○	○	○

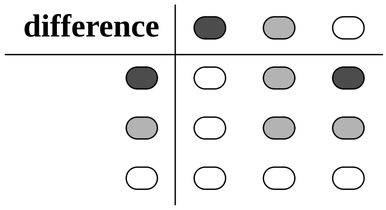
For example, the *union* of a kernel part with any other part is a kernel part since the union of two regions asks for membership in either region and since membership is certain for each kernel part. Likewise, the union of two boundaries or the union of a boundary with the outside should be a boundary, and only the parts of the space which belong to the outside of both regions contribute to the outside of the union.

On the other hand, the outside of the *intersection* is given by either region's outside because intersection requires membership in both regions. The kernel of the intersection only contains parts which definitely belong to the kernel of both arguments, and intersections of boundary parts with each other or with kernel parts make up the boundary of the intersection.

The definition of difference is motivated by the definition of complement. Clearly, the complement of the kernel should be the outside, and the complement of the outside should be the kernel, but what about the boundary part? Anything inside the vague part of an object might or might not belong to the object, so we cannot definitely say that the complement of the vague part is the outside. Neither can we say that the complement belongs to the kernel. So the only reasonable definition is to define the complement of the boundary to be the boundary itself:

complement	●	◍	○
	○	◍	●

Now the result of removing a vague region *v* from another vague region *u* can be defined as the intersection of *u* with the complement of *v*. That is, removing a kernel part means intersection with the outside which always yields outside, and removing anything from the outside leaves the outside part unaffected. Similarly, removing a boundary means intersection with the boundary and thus results in a boundary for kernel and boundary parts, and removing the outside of *v* (that is, nothing) does not affect any part of *u*.



Next we formally define these operations simply by using regions operations, that is, we express the notion of vague regions using well-understood exact regions. Let *u* and *v* be two vague regions. Then we define:

u **union** *v* := $(u^\kappa \oplus v^\kappa, (u^\beta \oplus v^\beta) \ominus (u^\kappa \oplus v^\kappa))$
u **intersection** *v* := $(u^\kappa \otimes v^\kappa, (u^\beta \otimes v^\beta) \oplus (u^\kappa \otimes v^\beta) \oplus (u^\beta \otimes v^\kappa))$
u **difference** *v* := $(u^\kappa \otimes (\ominus v^\kappa), (u^\beta \otimes v^\beta) \oplus (u^\kappa \otimes v^\beta) \oplus (u^\beta \otimes (\ominus v^\kappa)))$
complement *v* := $(\ominus v^\kappa, v^\beta)$

In the following we use as an abbreviating notation for the intersection of two (determinate) regions simple juxtaposition, and we assign intersection higher associativity than union and difference. That is, the above definition for *u* **difference** *v* could also be written more concisely as $(u^\kappa(\ominus v^\kappa), u^\beta v^\beta \oplus u^\kappa v^\beta \oplus u^\beta(\ominus v^\kappa))$.

It is not difficult to check that the definitions realize the behavior specified by the tables given above. Consider, for example, the **union**-operation. For $w = u$ **union** v we have to show the following three identities:

(1) $w^\kappa = u^\kappa v^\kappa \oplus u^\kappa v^\beta \oplus u^\kappa v^\varepsilon \oplus u^\beta v^\kappa \oplus u^\varepsilon v^\kappa$
(2) $w^\beta = u^\beta v^\beta \oplus u^\beta v^\varepsilon \oplus u^\varepsilon v^\beta$
(3) $w^\varepsilon = u^\varepsilon v^\varepsilon$

For proving (1) we first observe that $\oplus$ is idempotent. We can therefore duplicate the first term $u^\kappa v^\kappa$. Then using the fact that $\otimes$ distributes over $\oplus$ we can factorize both u^κ and v^κ and obtain:

$$w^\kappa = (u^\kappa(v^\kappa \oplus v^\beta \oplus v^\varepsilon)) \oplus (v^\kappa(u^\kappa \oplus u^\beta \oplus u^\varepsilon))$$

Since $v^\kappa \oplus v^\beta \oplus v^\varepsilon = \mathbf{1}_R$ and $u^\kappa \oplus u^\beta \oplus u^\varepsilon = \mathbf{1}_R$ and since $\mathbf{1}_R$ is the identity of $\otimes$ we get

$$w^\kappa = (u^\kappa \otimes \mathbf{1}_R) \oplus (v^\kappa \otimes \mathbf{1}_R) = u^\kappa \oplus v^\kappa,$$

which is the definition of the kernel part of **union**. Equation (2) can be shown as follows. For arbitrary regions *r* and *s* we know:

$$r \oplus s = rs \oplus r(\ominus s) \oplus (\ominus r)s$$

We can use this identity to rewrite the boundary definition as:

$$u^\beta v^\beta \oplus u^\beta(\ominus v^\beta) \oplus (\ominus u^\beta)v^\beta \ominus (u^\kappa v^\kappa \oplus u^\kappa(\ominus v^\kappa) \oplus (\ominus u^\kappa)v^\kappa)$$

Next we evaluate all complements (note that $\ominus v^\beta = v^\kappa \oplus v^\varepsilon$ or $\ominus v^\kappa = v^\beta \oplus v^\varepsilon$):

$$u^\beta v^\beta \oplus u^\beta(v^\kappa \oplus v^\varepsilon) \oplus (u^\kappa \oplus u^\varepsilon)v^\beta \ominus (u^\kappa v^\kappa \oplus u^\kappa(v^\beta \oplus v^\varepsilon) \oplus (u^\beta \oplus u^\varepsilon)v^\kappa)),$$

and apply distributivity of $\otimes$:

$$u^\beta v^\beta \oplus u^\beta v^\kappa \oplus u^\beta v^\varepsilon \oplus u^\kappa v^\beta \oplus u^\varepsilon v^\beta \ominus (u^\kappa v^\kappa \oplus u^\kappa v^\beta \oplus u^\kappa v^\varepsilon \oplus u^\beta v^\kappa \oplus u^\varepsilon v^\kappa)$$

In the resulting term, only $u^\beta v^\kappa$ and $u^\kappa v^\beta$ appear in both parts of the difference; all other intersections to be subtracted have no effect at all since all intersections are pairwise disjoint. Therefore the result is:

$$u^\beta v^\beta \oplus u^\beta v^\varepsilon \oplus u^\varepsilon v^\beta$$

which is exactly the condition required for w^β. For the proof of relationship (3), first note that in a boolean lattice we have for any two regions r and s: $\mathbf{1}_R \otimes s = s$, $\mathbf{1}_R \oplus s = \mathbf{1}_R$, and $\mathbf{1}_R = r \oplus (\ominus r)$. Therefore, we know that $s = (r \oplus (\ominus r))s = rs \oplus (\ominus r)s$, and it follows that $r \oplus s = r \oplus rs \oplus (\ominus r)s$. We also know that $r \oplus rs = r(\mathbf{1}_R \oplus s) = r$, so that $r \oplus s = r \oplus (\ominus r)s$. Since $(\ominus r)s$ is another way of denoting the difference $s \ominus r$, we get: $r \oplus (s \ominus r) = r \oplus s$. Now we have by definition that

$$w^\varepsilon = \ominus(w^\kappa \oplus w^\beta) = \ominus(u^\kappa \oplus v^\kappa \oplus ((u^\beta \oplus v^\beta) \ominus (u^\kappa \oplus v^\kappa))) = \ominus(u^\kappa \oplus v^\kappa \oplus u^\beta \oplus v^\beta)$$

By commutativity and de Morgan's law this reduces to:

$$\ominus(u^\kappa \oplus u^\beta) \otimes (\ominus(v^\kappa \oplus v^\beta))$$

which is by the definition of complement equal to $u^\varepsilon \otimes v^\varepsilon$, the condition required for w^ε. The correctness of the other operations is shown in a similar way.

In addition to having the four basic spatial operations on vague regions, it is also sometimes helpful to be able to explicitly deal with their boundary and kernel parts. Thus, we define the following operations:

boundary(v) := $(\varnothing, v^\beta)$
kernel(v) := $(v^\kappa, \varnothing)$
invert(v) := (v^β, v^κ)

In particular, these operations facilitate the computation with parts of vague regions in a purely exact way since the vague spatial operations, applied to vague regions with an empty boundary, behave exactly like the corresponding exact spatial operations. (This can be easily seen from the definitions.)

4.2 Vague Predicates

One of the most basic relationships that can be observed for two regions is whether they intersect or not. Many different cases of intersection can be identified leading to specialized predicates, like *covers* or *meets*, that describe more specific relationships. To define an intersection predicate for two vague regions u and v it is instructive to look at the possible results for the kernel and boundary of $w = u$ **intersection** v. Surely, we want to say that u and v intersect if $w^\kappa = u^\kappa v^\kappa$ is not empty, that is, if the kernel regions of u and v overlap. This is true independent from the value of w^β. Likewise, if the regions of $u^\kappa \oplus u^\beta$ and $v^\kappa \oplus v^\beta$ are disjoint, we can safely say that u and v do not intersect at all. However, if $w^\kappa = \mathbf{0}_R$ and $w^\beta \neq \mathbf{0}_R$, we cannot be sure about the intersection of u and v. This means, we can neither return *true* nor *false*, but we rather have to define the predicate to yield something like *maybe* or *unknown* (comparable to NULL-values known from relational databases).

Therefore, we use a three-valued logic as the range of boolean predicates. The definition of the logical operators parallels the definition of the operations for vague regions (1, 0, and ? are used as abbreviations for *true*, *false*, and *maybe*):

and	1	?	0
1	1	?	0
?	?	?	0
0	0	0	0

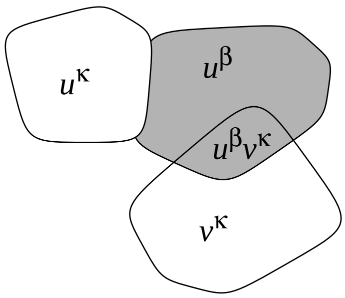
or	1	?	0
1	1	1	1
?	1	?	?
0	1	?	0

not	1	?	0
	0	?	1

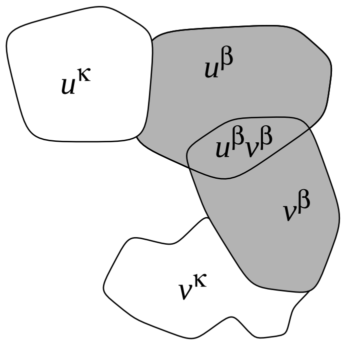
Now we return to the definition of vague predicates. For example, the definition of intersection is:

$$u \textbf{ intersects } v = \begin{cases} \textit{true} & \text{if } u^\kappa v^\kappa \neq \mathbf{0}_R \\ \textit{false} & \text{if } u^\kappa v^\kappa \oplus u^\beta v^\beta \oplus u^\kappa v^\beta \oplus u^\beta v^\kappa = \mathbf{0}_R \\ \textit{maybe} & \text{otherwise} \end{cases}$$

The *maybe*-case of intersection can be distinguished further according to whether a kernel/boundary or only a boundary/boundary intersection exists. An example for both situations is shown below:



vague intersection



weak vague intersection

We consider the situation depicted on the left to be a stronger indication of intersection than the situation on the right. Accordingly, we define two predicates, **v-intersects** (*vague intersection*) and **w-intersects** (*weak vague intersection*) as follows:

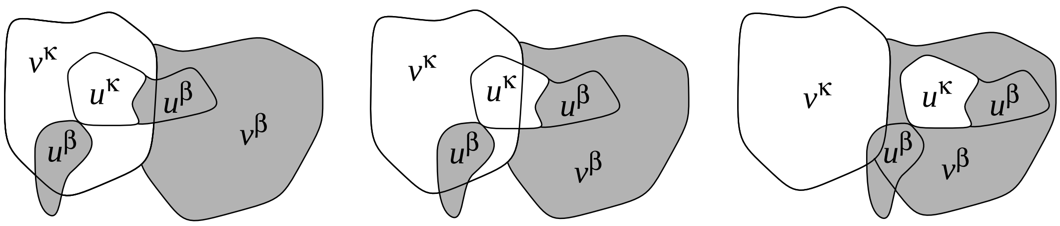
$$u \textbf{ v-intersects } v = \begin{cases} \textit{true} & \text{if } u^\beta v^\kappa \oplus u^\kappa v^\beta \neq \mathbf{0}_R \text{ and } u^\kappa v^\kappa = \mathbf{0}_R \\ \textit{false} & \text{otherwise} \end{cases}$$

$$u \textbf{ w-intersects } v = \begin{cases} \textit{true} & \text{if } u^\beta v^\beta \neq \mathbf{0}_R \text{ and } u^\beta v^\kappa \oplus u^\kappa v^\beta \oplus u^\kappa v^\kappa = \mathbf{0}_R \\ \textit{false} & \text{otherwise} \end{cases}$$

A special case of intersection is also given when *u* lies inside *v*. We can safely say that *u* **inside** *v* holds if everything of *u* (that is, kernel and boundary) is inside the kernel of *v*. If this is not the case, we cannot simply conclude that *u* **inside** *v* is *false* since this requires definite knowledge about a part of *u* being outside any part of *v*. In other words, whenever $u^\kappa \subseteq v^\kappa \oplus v^\beta$ we are not sure about insideness, and we should define *u* **inside** *v* as *maybe*:

$$u \textbf{ inside } v = \begin{cases} \textit{true} & \text{if } u^\kappa \oplus u^\beta \subseteq v^\kappa \\ \textit{false} & \text{if } u^\kappa \not\subseteq v^\kappa \oplus v^\beta \\ \textit{maybe} & \text{otherwise} \end{cases}$$

As we have done for intersection we can discriminate the *maybe*-case further. If the kernel part of *u* is completely inside the kernel part of *v*, then only the boundary of *u* makes the decision of insideness vague. This is a stronger indication for the inside relationship than in the case that also a part of *u*'s kernel lies in the boundary of *v*. Some possible relationships are shown in the following picture:



vague inside *weak vague inside*

As indicated by the two situations on the right, the predicate for *weak vague inside* can be distinguished further. We do not follow this line, since this leads quickly to an inflation of predicates.

$$u \textbf{ v-inside } v = \begin{cases} \textit{true} & \text{if } u^\kappa \subseteq v^\kappa \text{ and } u^\beta \not\subseteq v^\kappa \\ \textit{false} & \text{otherwise} \end{cases}$$

$$u \textbf{ w-inside } v = \begin{cases} \textit{true} & \text{if } u^\kappa \subseteq v^\kappa \oplus v^\beta \text{ and } u^\kappa \not\subseteq v^\kappa \\ \textit{false} & \text{otherwise} \end{cases}$$

The complementary predicate for **intersects** is **disjoint**, and its definition is obtained by simply exchanging *true* and *false* in the definition of **intersects**.

Note that we cannot directly express relationships, such as *meets* or *adjacent*, since we currently have no concept of lines and points in our model. However, we can regard weak vague intersections as a kind of adjacency as done by [CF96, CG96].

4.3 Vague Numeric Operations

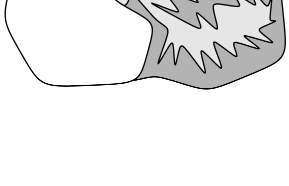
The definition of numeric operations on vague regions must be based on the corresponding functions for regions. Let us consider, for example, the *area* of a vague region. The area is at least the area of the kernel part and at most the area of kernel plus boundary. So the result of the vague area operation could be an interval given by the minimum and maximum area values. We then, however, have to work with intervals in any further calculations using such an area value. This requires a whole new set of vague arithmetic operations working with intervals. (The situation is similar to the extension to three-valued logic used for predicates.) So in order to keep things simple we instead define *two* operations, **min-area** and **max-area**, and can thus keep ordinary numeric operations.

min-area(v) := *area* (v^{κ})
max-area(v) := *area* ($v^{\kappa} \oplus v^{\beta}$)

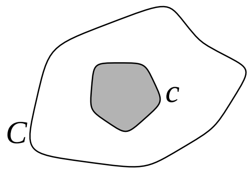
The definition of the *distance* between two vague regions u and v is very similar. Again the distance is a vague value: An upper bound is obtained by the distance between the kernel parts of u and v, that is, we are sure that the distance is at most the distance between the kernels. The distance might be smaller, but it is at least as large as the distance between the maximal extensions of u and v, in other words, the minimum distance is given by the distance taking kernel and boundary into account.

min-dist (u, v) := *dist* ($u^{\kappa} \oplus u^{\beta}$, $v^{\kappa} \oplus v^{\beta}$)
max-dist (u, v) := *dist* (u^{κ}, v^{κ})

The generalization of *area* and *dist* to vague regions is rather straightforward. There are other useful operations on regions, however, for which a generalization to the vague case is not quite so simple or even impossible. Consider, for example, the definition of perimeter. The definition for the exact case is well-known, but what could be the perimeter of a vague region? In a first approach one could be tempted to define minimum and maximum versions similar to the definition of area. This, however, might lead to wrong results. We have indicated that the boundary region can be thought of (at least in some applications) as describing possible locations of the region's contour. But then we cannot give any upper bound on the length of such a curve. In particular, the contour might be much longer than the perimeter of the boundary region, for instance:

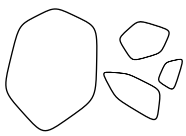


Moreover, we cannot simply take the perimeter of the kernel part as the minimal perimeter. This can be seen as follows. Usually, holes contribute to the perimeter of a region. If now, for example, a kernel part of a vague region *v* contains a hole which is equal to a boundary part of *v*, the perimeter of the hole is not counted in the perimeter of the maximal possible region.

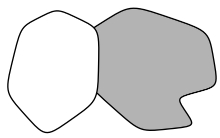


In this example, the minimal possible region has a perimeter of *length*(*C*) + *length*(*c*) whereas the maximal possible region has the minimal possible perimeter of *length*(*C*).

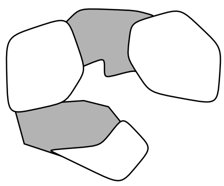
Another example is an operation giving the number of connected components for which the generalization to vague regions heavily depends on the semantics of the boundary parts. Since regions need not be connected, a "possible regions"-semantics of boundary parts might well allow several unconnected parts, that is,



could be a possible region of



Hence, we cannot give an upper bound on the number of components. But, in general, we cannot even give a non-trivial lower bound either (for example, the number of kernel components) since kernel components might be connected by boundary regions. Thus in the example below, the minimal number of components is 1 although there are three kernel regions.



(If we required, however, that a possible region extending into an adjacent boundary is connected, we could give a meaningful definition.)

We deliberately have avoided definitions of operations, such as perimeter and number of components, since we are then not forced to fix a semantics for vague regions. This means, the semantics can be assumed by each application as required and thus makes our model more general.

5 Embedding Vague Regions into Query Languages

In the previous section we have defined operations on vague regions. Next we indicate how these operations can play a part in a spatial query language. We do not give a full description of a specific language. We rather assume a relational data model where tables may contain vague region objects together with a SQL-like query language. For example, if we want to find out all regions where lack of water is a problem for cultivation, we can pose the following query:

select region **from** weather **where** climate = dry

Here we assume a table *weather* having a column named *region* containing vague region values for various climatic conditions given by the column *climate*. A similar query could ask for bad soil regions as a hindrance for cultivation.

Note that the result of both queries is a *set* of vague regions. If we now want to find out about regions where cultivation is impossible due to either reason, we ask for the union of the two region sets. Thus, we first have to cast the sets into single region objects. We therefore use the built-in aggregation function **sum** which, when applied to a set of regions, aggregates this set by repeated application of **union** (in the sense of *fold*/*reduce* of functional languages). So we can determine regions where cultivation is impossible by:

(**select sum**(region) **from** weather **where** climate = dry)
union
(**select sum**(region) **from** soil **where** quality = bad)

Pollutions are nowadays a central ecological problem and cause an increasing number of environmental damages. Important examples are air pollution and oil soiling. Pollution control institutions, ecological researchers, and geographers, usually use maps for visualizing the expansion of pollution. We can ask, for example, for inhabitable areas which are air polluted (where the kernel part of air pollution denotes heavily polluted areas and the boundary part gives only slightly polluted regions).

select **sum**(pollution.region) **intersection sum**(areas.region)
from pollution, areas
where area.use = inhabited **and** pollution.type = air

Then the kernel part of the result consists of inhabited regions which are heavily polluted, and the boundary consists (a) of slightly polluted inhabited regions, (b) of heavily polluted regions which are only partially inhabited, and (c) of slightly polluted and partially inhabited regions. If we want to reach all people who live in heavily polluted areas, we need the kernel of the intersection together with part (b) of the intersection boundary. How can we get this from the above query? The trick is to force boundary parts (a) and (c) to be empty by restricting pollution areas to their kernel region:

select kernel(**sum**(pollution.region)) **intersection sum**(areas.region)
from …

A slightly different query is to find out all areas where people are definitely or possibly endangered by pollution. Of course, we have to use an intersection predicate. More precisely, we want to find those areas for which **intersects** either yields *true* or *maybe*. For this purpose we can prefix any predicate with **maybe** which causes the predicate to fail only if it returns *false*. (Technically **maybe** turns a *maybe* value into *true*.) So the query is:

select areas.name
from pollution, areas
where area.use = inhabited **and**
pollution.region **maybe intersects** areas.region

We could also express the query by using simply **intersects** and explicitly adding the two cases for **v-intersects** and **w-intersects**. This would be, of course, much longer and less clear.

The following example describes a situation which stresses the conflicting interests of economy and ecology. Assume on the one hand areas of animal species and plants that are worth being protected (nature reserves and national parks are the kernel regions) and on the other hand mineral resources the mining of which prospects high profits. An example for forming a difference of vague regions is a query which asks for mining areas that do not affect the living space of endangered species.

(**select sum**(region) **from** resources **where** kind = mineral)
difference
(**select sum**(region) **from** nature **where** type = endangered)

The kernel of the result describes regions where mining should be allowed. The boundary consists (a) of regions where mineral resources are uncertain and (b) of resource kernels that lie in (non-kernel) regions hosting endangered species. Since national parks are generally protected by the government, it is especially regions (b) conservationists should carefully observe. We can determine these regions by:

(**select kernel**(**sum**(region)) **from** resources **where** kind = mineral)
intersection
(**select boundary**(**sum**(region)) **from** nature **where** type = endangered)

The result is a vague region with an empty kernel and a boundary that just consists of the intersection of the mineral resource kernel and the endangered nature boundary.

Next we consider an example from biology already mentioned in Section 3. Assume we are given living spaces of different animal species. The kernel describes places where they normally live, and the boundary describes regions where they can be found occasionally (for example, to hunt for food or to migrate from one kernel area to another through a corridor). First, we can search for pairs of species which share a common living space. This asks for regions which have a non-empty intersection kernel:

select A.name, B.name
from animals (A), animals (B)
where A.region **intersects** B.region

A quite different question, also based on intersection, is whether there are animals that only sometimes enter the kernel region of other animals, for example, to attack them (but usually live in different areas). Here, we ask for an empty intersection kernel and a non-empty boundary/kernel intersection which is exactly the concept of vague intersection. The above query changes to:

select …
where A.region **v-intersects** B.region

Finally, we can also ask for animals that only encounter each other in their boundaries. This would be, for instance, the case for two animal species that are both hunters and usually avoid contact. The corresponding query is obtained by simply using **w-intersects** instead of **v-intersects**.

Assume that we are given a map of land areas (kernels) and mixed areas like shores and banks (boundaries) where the living spaces of animals with their kernel and vague regions are depicted. We can ask for animals that usually live on land and sometimes enter the water or for species that never leave their land area. This can be expressed using the inside predicate. The first example is characterized by an animal's living space being **v-inside** land:

select name
from animals
where region **v-inside** (**select sum**(region) **from** land)

The second example demands plain (that is, strong) **inside**.

A quite different example using insideness relates to the historical development of the Roman Empire, in particular, its expansion. At any moment during this development there were kernel regions representing the areas currently occupied by the Roman conquerors and spheres of influence (vague parts) that were under the control of the Roman Empire but not annexed. If we consider two points in time ($t_1 < t_2$) and hence two vague regions u and v, we could ask whether any of the occupied areas at t_1 have been preserved to t_2. This is the case if the kernel of u is completely inside v, in other words, when u **inside** v **or** u **v-inside** v. If they had to give up kernel regions, this is an example of u **w-inside** v.

Let us finally provide some examples for numeric operations. Oil companies are often interested to determine whether it is worth exploiting a recently discovered oilfield. Hence, they classify oilfields in areas where the existence of oil was proved by soil samples and in areas where the incidence of oil is only assumed. The decision of exploiting then depends on the guaranteed minimal extensions of the oilfields:

select min-area(region) **from** oilfields

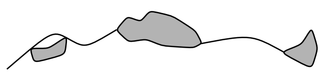
An example for applying **max-area** is again pollution where we should be pessimistic and consider the worst case of all possible polluted regions.

The minimum distance between a forest fire and a region of endangered species indicates where protective measures should be performed first. For another example consider an attacked country that might have secure parts (kernel), battle regions (boundary), and even lost parts (outside). To move from one safe area to another one might consider the risk of such a trip be given by the maximum distance between different regions, that is, the difference between secure parts.

6 Conclusions and Future Work

We have defined a data model of regions that is capable of describing many different aspects of vague spatial objects. It is a canonical extension of a determinate region model which facilitates the treatment of vague and exact regions in one model. In particular, this allows a smooth migration from already existing models to vague concepts (at least as far as regions are concerned). Our approach is based on exact spatial modeling concepts which allows to build upon existing work and simplifies many definitions. In particular, we can (re-)use already existing regions implementations to realize vague regions with only minimal effort.

Of course, the current model is limited in some ways, and we are currently investigating extensions along several different lines. First, the presented concept of vagueness can be extended to *other spatial objects*, such as points and lines. For example, a vague line could be thought of as consisting of a kernel part given by a set of (unconnected) curves and a vague part described by a boundary region. An example is a river which may contain fixed segments (determined, for instance, by levees) and a boundary which describes possible flows that depend on water level or season.



A vague point can be simply given by a vague region (with empty kernel) describing possible positions of the point. To define such extensions we first have to extend the basic model of exact regions by lines and points together with operations defined for them. These can then be used to define vague lines and points. We also should consider operations concerning objects of different vague types, for example, the intersection of a vague line with a vague region.

Another direction of extension is the notion of vagueness itself. As yet, there is only one kind of vagueness, but there are many applications which can be best described by having *different degrees of vagueness*. For example, zones of decreasing pollution or regions of different possibilities for certain virus infections. Our model can be easily extended to deal with this kind of applications by having a set of regions labeled with different values of a suitable domain *D* which is subject to certain restrictions. For example, we need operations *max* and *min* to give meaningful extensions of operations like **union** and **intersection**.

Finally, we consider the *integration* of vague regions (and their possible extensions) into other data models. We have already seen how for predicates and numeric operations the vagueness of regions affects the corresponding domains of booleans and real numbers. It is likely that the situation is similar for other domains as well. So the integration of vague regions into any existing data model and query language might cause some trouble since it either requires a redefinition of the data types or a redefinition (and duplication) of operations. That this can be tedious and error-prone has been demonstrated in the description of numeric operations. Note that the problem of "vague infection" is not restricted to standard data types. For example, in [EG94, Er94] graphs have been integrated into a spatial data model. With respect to vague spatial objects, an operation like **subgraph** that computes part of a graph according to a possibly spatial predicate should return a *vague graph*. Now, what are vague graphs, and how can all the graph operations adapted to the vague case? We currently consider the integration an open problem.

References

[Al61] P. Alexandroff. *Elementary Concepts of Topology.* Dover Publications, 1961.

[Al94] D. Altman. Fuzzy Set Theoretic Approaches for Handling Imprecision in Spatial Analysis. *Int. Journal of Geographical Information Systems*, vol. 8, no. 3, pp. 271-289, 1994.

[Ar83] M.A. Armstrong. *Basic Topology.* Springer Verlag, 1983.

[Ba93] R. Banai. Fuzziness in Geographical Information Systems: Contributions from the Analytic Hierarchy Process. *Int. Journal of Geographical Information Systems*, vol. 7, no. 4, pp. 315-329, 1993.

[Bl84] M. Blakemore. Generalization and Error in Spatial Databases. *Cartographica*, vol. 21, 1984.

[Bu96] P.A. Burrough. Natural Objects with Indeterminate Boundaries. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series, vol. 3, Taylor & Francis, pp. 3-28, 1996.

[CF96] E. Clementini & P. di Felice. An Algebraic Model for Spatial Objects with Indeterminate Boundaries. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series, vol. 3, Taylor & Francis, pp. 153-169, 1996.

[CG96] A.G. Cohn & N.M. Gotts. The 'Egg-Yolk' Representation of Regions with Indeterminate Boundaries. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series, vol. 3, Taylor & Francis, pp. 171-187, 1996.

[Co96] H. Couclelis. Towards an Operational Typology of Geographic Entities with Ill-defined Boundaries. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series, vol. 3, Taylor & Francis, pp. 45-55, 1996.

[ECF94] M.J. Egenhofer, E. Clementini & P. di Felice. Topological Relations between Regions with Holes. *Int. Journal of Geographical Information Systems*, vol. 8, no. 2, pp. 129-142, 1994.

[Ed94] G. Edwards. Characterizing and Maintaining Polygons with Fuzzy Boundaries in GIS. *6th Int. Symp. on Spatial Data Handling*, pp. 223-239, 1994.

[Eg89] M.J. Egenhofer. Spatial SQL: A Spatial Query Language. Report 103, Dept. of Surveying Engineering, University of Maine, 1989.

[EG94] M. Erwig & R.H. Güting. Explicit Graphs in a Functional Model for Spatial Databases. *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, no. 5, pp. 787-804, 1994.

[Er94] M. Erwig. *Graphs in Spatial Databases*. Doctoral Thesis, FernUniversität Hagen, 1994.

[Fi93] J.T. Finn. Use of the Average Mutual Information Index in Evaluating Classification Error and Consistency. *Int. Journal of Geographical Information Systems*, vol. 7, no. 4, pp. 349-366, 1993.

[Ga64] S. Gaal. *Point Set Topology*. Academic Press, 1964.

[GS93] R.H. Güting & M. Schneider. Realms: A Foundation for Spatial Data Types in Database Systems. *3rd Int. Symp. on Large Spatial Databases*, pp. 14-35, 1993.

[GS95] R.H. Güting & M. Schneider. Realm-Based Spatial Data Types: The ROSE Algebra. *VLDB Journal*, vol.4, pp. 100-143, 1995.

[Gü88] R.H. Güting. Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems. *Int. Conf. on Extending Database Technology*, pp. 506-527, 1988.

[KV91] V.J. Kollias & A. Voliotis. Fuzzy Reasoning in the Development of Geographical Information Systems. *Int. Journal of Geographical Information Systems*, vol. 5, no. 2, pp. 209-223, 1991.

[LAB96] P. Lagacherie, P. Andrieux & R. Bouzigues. Fuzziness and Uncertainty of Soil Boundaries: From Reality to Coding in GIS. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series, vol. 3, Taylor & Francis, pp. 275-286, 1996.

[LN87] U. Lipeck & K. Neumann. Modelling and Manipulating Objects in Geoscientific Databases. *5th Int. Conf. on the Entity-Relationship Approach*, pp. 67-86, 1987.

[Sc95] M. Schneider. *Spatial Data Types for Database Systems*. Doctoral Thesis, FernUniversität Hagen, 1995.

[Sc96] M. Schneider. Modelling Spatial Objects with Undetermined Boundaries Using the Realm/ROSE Approach. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series, vol. 3, Taylor & Francis, pp. 141-152, 1996.

[Sh93] R. Shibasaki. A Framework for Handling Geometric Data with Positional Uncertainty in a GIS Environment. *GIS: Technology and Applications*, World Scientific, pp. 21-35, 1993.

[SV89] M. Scholl & A. Voisard. Thematic Map Modeling. *1st Int. Symp. on Large Spatial Databases*, pp. 167-190, 1989.

[Ti80] R.B. Tilove. Set Membership Classification: A Unified Approach to Geometric Intersection Problems. *IEEE Transactions on Computers*, vol. C-29, pp. 874-883, 1980.

[Us96] E. L. Usery. A Conceptual Framework and Fuzzy Set Implementation for Geographic Features. *Geographic Objects with Indeterminate Boundaries*, GISDATA Series, vol. 3, Taylor & Francis, pp. 71-85, 1996.

[Wa94] F. Wang. Towards a Natural Language User Interface: An Approach of Fuzzy Query. *Int. Journal of Geographical Information Systems*, vol. 8, no. 2, pp. 143-162, 1994.

[WB93] M.F. Worboys & P. Bofakos. A Canonical Model for a Class of Areal Spatial Objects. *3rd Int. Symp. on Advances in Spatial Databases*, Springer-Verlag, LNCS 692, pp. 36-52, 1993.

[WH96] F. Wang & G.B. Hall. Fuzzy Representation of Geographical Boundaries in GIS. *Int. Journal of Geographical Information Systems*, vol. 10, no. 5, pp. 573-590, 1996.

[WHS90] F. Wang, G.B. Hall & Subaryono. Fuzzy Information Representation and Processing in Conventional GIS Software: Database Design and Application. *Int. Journal of Geographical Information Systems*, vol. 8, no. 2, pp. 143-162, 1994.

[Za65] L.A. Zadeh. Fuzzy Sets. *Information and Control*, vol. 8, pp. 338-353, 1965.

Appendix

We give a formal definition of the general model for determinate regions that has been informally described in Section 3. An adequate and general method to formally define this model is to use the point set paradigm and point set topology. The point set paradigm expresses that space is composed of infinitely many points and that spatial objects like areal objects are distinguished subsets of space which are viewed as entities. Point set topology [Al61, Ar83, Ga64] allows one to distinguish special topological structures of a point set like its boundary or interior. We start with some basic concepts of point set topology.

Definition. Let X be a set and $T \subseteq 2^X$ be a subset of the power set of X. The pair (X, T) is called a *topological space*, if the following three axioms are satisfied:

(T1) $X \in T, \varnothing \in T$

(T2) $U \in T, V \in T \Rightarrow U \cap V \in T$

(T3) $S \subseteq T \Rightarrow \bigcup_{U \in S} U \in T$

T is called a *topology* for X. The elements of T are called *open sets*, their complements in X *closed sets*. The elements of X are called *points*.

When no confusion can arise, T is not mentioned, and X denotes a topological space. In the sequel, let X be a topological space and $Y \subseteq X$.

Definition. The *interior* of Y, denoted by Y°, is the union of all open sets that are contained in Y. The *closure* of Y, denoted by $\overline{Y}$, is the intersection of all closed sets that contain Y. The *exterior* of Y, denoted by Y^-, is the union of all open sets that are not contained in Y. The *boundary* of Y, denoted by ∂Y, is the intersection of the closure of Y and the closure of the complement of Y, that is, $\partial Y = \overline{Y} \cap \overline{X - Y}$.

The relationships between these four topological structures are given by the provable statements (1) $Y^\circ \cap \partial Y = \varnothing$, (2) $Y^\circ \cup \partial Y = \overline{Y}$, (3) $Y^- \cap \partial Y = \varnothing$, and (4) $Y^\circ \cap Y^- = \varnothing$. Obviously we can conclude $X = \partial Y \cup Y^\circ \cup Y^-$.

Since our objective is to model two-dimensional areal objects for spatial applications, we embed them in the Euclidean space (plane) $\mathbb{R}^2$ as an instance of a topological space[4] with metric properties. A problem of applying pure set-theoretic operations to point sets is that undesired geometric anomalies can arise. These anomalies are avoided by the concept of *regularity* [Ti80].

Definition. Y is called *regular closed* if $Y = \overline{Y^\circ}$.

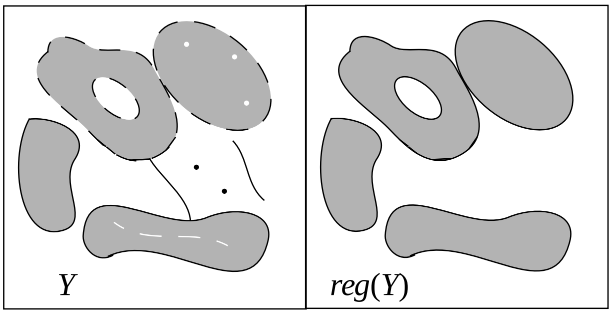
Intuitively, regular closed sets model areal objects containing their boundaries and avoid both isolated or dangling line or point features and missing lines and points in the form of cuts and punctures. Hence, it makes sense to define a *regularization* function *reg* which associates a set Y with a regular closed set, as follows:

---

4. Note that most of the definitions and results in the sequel also hold for general topological spaces.

$reg(Y) := \overline{Y^{\circ}}$

An example of regularization is shown below where the set Y consists of areal, point, and line objects. Some areal objects contain only parts of their boundaries (drawn with broken lines) and have cuts (drawn with broken lines) and punctures. The regularization process eliminates point and line features, cuts and punctures, and includes the missing boundary parts of the areal objects.



The union of a finite number of regular closed sets is regular closed. The intersection and difference of regular closed sets are not necessarily regular closed. Hence, we introduce *regular set operations* that preserve regularity.

Definition. Let A, B be regular closed sets, and let $\neg A$ denote the (set-theoretic) complement $\mathbb{R}^2 - A$ of A. Then

(i) $A \cup_r B := reg(A \cup B) = A \cup B$
(ii) $A \cap_r B := reg(A \cap B)$
(iii) $A -_r B := reg(A - B)$
(iv) $\neg_r A := reg(\neg A)$

It is obvious that the subspace *RCS* of regular closed sets together with the regular set operations is a topological space. Regular closed sets and regular set operations express a natural formalization of the dimension-preserving property taken for granted by many spatial type systems and geometric algorithms. The following important theorem holds:

Theorem. *RCS* with the set-theoretic order relation $\subseteq$ is a Boolean lattice.

This implies that (i) (RCS, $\subseteq$) is a partially ordered set, (ii) every pair A, B of elements of *RCS* has a least upper bound $A \cup_r B$ and a greatest lower bound $A \cap_r B$, (iii) (RCS, $\subseteq$) has a maximal element $\mathbf{1}_r := \mathbb{R}^2$ (identity of $\cap_r$) and a minimal element $\mathbf{0}_r := \varnothing$ (identity of $\cup_r$), (iv) algebraic laws like idempotence, commutativity, associativity, and distributivity hold for $\cup_r$ and $\cap_r$, (v) (RCS, $\subseteq$) is a complementary lattice, that is, $\forall A \in RCS : A \cap_r \neg_r A = \mathbf{0}_r$ and $A \cup_r \neg_r A = \mathbf{1}_r$.

Definition. A *region* is a regular closed set.

Definition. The type R consists of all regions and has the operations $\oplus$, $\otimes$, $\ominus$, and $\circleddash$ that are equated with the regular set operations $\cup_r$, $\cap_r$, $-_r$, and $\neg_r$, respectively, and the elements $\mathbf{1}_R = \mathbf{1}_r$ and $\mathbf{0}_R = \mathbf{0}_r$.