Functional Extensions of a Raster Representation for Topological Relations

Stephan Winter and Andrew U. Frank

Dept. of Geoinformation
Technical University Vienna
Gusshausstr. 27-29, 1040 Vienna, Austria
{winter,frank}@geoinfo.tuwien.ac.at
http://www.geoinfo.tuwien.ac.at

Abstract. Topological relations are not well defined for raster representations. In particular the widely used classification of topological relations based on the nine-intersection [8, 5] cannot be applied to raster representations [9]. But a raster representation can be completed with edges and corners [14] to become a cell complex with the usual topological relations [16]. Although it is fascinating to abolish some conceptual differences between vector and raster, such a model appeared as of theoretical interest only.

In this paper definitions for topological relations on a raster – using the extended model – are given and systematically transformed to *functions* which can be applied to a regular raster representation. The extended model is used only as a concept; it need not to be stored. It becomes thus possible to determine the topological relation between two regions, given in raster representation, with the same reasoning as in vector representations. This contributes to the merging of raster and vector operations. It demonstrates how the same conceptual operations can be used for both representations, thus hiding in one more instance the difference between them.

1 Introduction

Topological relations are not well defined for raster representations. In particular the widely used classification of topological relations based on the four- and nine-intersection [8, 5] cannot be applied to raster representations [9]. This is due to the topological incompleteness of a raster: it consists, in the field view, of (open) two-dimensional cells only. In contrast, vector representations consist also of one- and zero-dimensional elements, used for the representation of boundaries, which close two-dimensional point sets and demarcate from their exterior. Boundary constructions in raster representations require the use of raster elements [13], although they are two-dimensional by nature. Two-dimensional boundaries contradict to topology, so they cause some well-known paradoxes.

Kovalevsky has suggested that the raster can be completed with edges and nodes to become a full topological model [14]. In this representation, called

Winter, Stephan, and Andrew U. Frank. "Functional Extensions of a Raster Representation for Topological Relations." In *Interoperating Geographic Information Systems (Proceedings of Interop'99, Zurich)*, edited by Andrej Vckovski, Kurt E. Brassel and Hans-Jörg Schek, 293-304. Berlin: Springer-Verlag, 1999.

here a *hybrid* raster, topological relations are defined equivalent to a vector representation [16]. But the hybrid raster appeared as of theoretical interest only, mostly due to its additional and redundant memory requirements (see Section 2.3).

Here detailed definitions for topological relations on a raster – using the hybrid raster representation – are given and then systematically transformed to yield functions which can be used in a convolution operation applied to a regular raster representation. Hereby the hybrid raster is only used as a concept. It need not be stored and is only partially constructed during the execution of a determination of a topological relation. It becomes thus possible to determine the topological relationship of two regions, given in raster representation, by the four- or nine-intersection.

A formal approach is used to understand the structure and the theory of an extended raster representation and its application for topological relations. The specification is written in a functional language. Pure functional languages [1], like Gofer [11], provide a useful separation of specification and implementation [10]. With executable specifications, the result is a provable code – in syntax as well as with test cases – with a clear semantic. Furthermore, such a specification is basis for iterative optimization; e.g. the Gofer code published here[1] was optimized in several cycles of improvements. The value of such formal specifications is recognized more and more. So Dorenbeck and Egenhofer presented a formal specification of raster overlay, with a generalization for polygons [3]. We also specify an overlay, but of an extended raster, deriving the same behavior of raster and vector representations.

This contributes to the merging of raster and vector operations. It demonstrates how the same conceptual operations can be used for both representations, thus hiding in one more instance the difference between them.

The paper is structured as follows. In Section 2 previous work is collected, regarding topological relations between regions, and hybrid raster representation. In Section 3 the raster representation is extended to a hybrid raster, and the combination of two raster images is presented to determine a four-intersection. It is also discussed how to optimize computations. An example in Section 4 shows the advantage of an executable specification. Finally a discussion sums up the results and perspectives (Section 5).

2 Previous Work

2.1 Topological Relations

Egenhofer proposed a representation of topological relations between point sets, based on the intersection sets of such point sets [6, 7, 5]. Point sets in $\mathbb{R}^2$ refer to Euclidean topology, with the Euclidean distance as a metric. The metric is needed to define a boundary of (open) sets. Distinguishing the interior $\mathcal{X}^\circ$, the boundary $\partial\mathcal{X}$ and the exterior $\mathcal{X}^c$ of a point set X, two point sets $\mathcal{A}$ and $\mathcal{B}$ may

[1] The complete code is available at our web-page.

have nine intersection sets, which form a partition of the plane. For describing topological properties the size of the intersection sets is irrelevant, only being empty or not is characterizing.

For regular closed and singular connected sets – *simple* regions – even four intersection sets are sufficient, because the omitted five intersection sets do not vary. The sets can be ordered in a 2×2-array, the four-intersection **I4**:

$$\mathbf{I4} = \begin{pmatrix} \mathcal{A}^\circ \cap \mathcal{B}^\circ & \mathcal{A}^\circ \cap \partial\mathcal{B} \\ \partial\mathcal{A} \cap \mathcal{B}^\circ & \partial\mathcal{A} \cap \partial\mathcal{B} \end{pmatrix} \tag{1}$$

The *nine-intersection* contains the other five sets, too. – Eight relationships between two simple regions can be characterized using this schema (Table 1).

Table 1. The eight distinct four-intersections for simple regions, and the names of the characterized topological relations.

$\begin{pmatrix} \emptyset & \emptyset \\ \emptyset & \emptyset \end{pmatrix}$	$\begin{pmatrix} \emptyset & \emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & \emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$
DISJOINT	MEET	OVERLAP	EQUAL
$\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset \end{pmatrix}$	$\begin{pmatrix} \neg\emptyset & \emptyset \\ \neg\emptyset & \emptyset \end{pmatrix}$
COVER	COVEREDBY	CONTAIN	CONTAINEDBY

The found relationships were investigated and applied to spatial reasoning [4, 12], with the interest to speed up spatial queries in GIS or in AI. They are an important improvement of vector representations, which base on point sets in $\mathbb{R}^2$.

2.2 Topological Relations and Raster Representations

A raster representation is a two-dimensional array of elements with integer coordinates. Interpreting the raster elements as fields – instead of lattice points –, the raster is a regular subdivision of space into squares of equal size, *resels* (short form for 'raster elements'). – For the general principle it doesn't matter how the raster is implemented (see e.g. [15]). But a comparison to vector representations directly shows that only (open) two-dimensional elements exist, and one- and zero-dimensional elements are missed. Boundaries of regions cannot be defined by infinite balls as in the Euclidean space.

This problem was treated so far in two ways:

- omitting boundaries, having only regions as open sets, as it is done in region based reasoning methods (e.g. [2]);

– defining substitutes for one-dimensional boundaries, using raster elements and any arbitrary neighborhood definition [13].

The first solution allows the application of the nine-intersection only for its two-dimensional intersection sets, i.e. the region *interiors* and *exteriors*, which yields a subset of the relationships in Euclidean space [17]. The resulting four-intersection may not be mixed up with the four-intersection defined with boundaries (Eq. 1).

The second solution generates two-dimensional boundaries, resel chains or bands, either as interior boundaries or as exterior boundaries. Two-dimensional boundaries contradict to topology, so they cause some well-known paradoxes [13]. The result of intersecting the sets of interior, boundary and exterior raster elements depends heavily on the definition of the boundary (interior or exterior). Even worse is the possibility of more than the eight four-intersections described in Table 1, simple regions presumed [9]. These intersections have no common-sense meaning; they appear as variations of the eight presented intersections and need special care.

2.3 The Hybrid Raster Representation

Kovalevsky has suggested that the raster can be completed with edges and nodes to become a full topological model, to be precisely: an abstract cell complex [14]. The only specialty of this cell complex is its regular structure (Figure 1). Generally all elements of a cell complex are called (2D-, 1D-, 0D-)cells, but we will speak in the following of two-dimensional *cells* – identical with the resels in raster –, one-dimensional *edges* and zero-dimensional *nodes*. The union of edges and nodes will be called the *skeleton* of the cells.

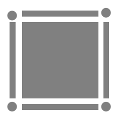

Fig. 1. A (regular shaped) cell complex, replacing a raster element of usual raster representations in the hybrid raster: each cell is closed by four edges and four nodes.

In this representation, topological relations regard again to Euclidean space, and the four- or nine-intersection can be applied in full accordance to vector representations [16]. In vector representations these tests are expensive, requiring polygon intersection. In a hybrid raster representation the tests are simple to evaluate: two hybrid rasters (of the same resolution, same size and common origin), labeled by three values for interior, boundary and exterior, are overlaid by $\wedge$ (equivalent to $\cap$ in set denotation). Then the nine possible combinations can be accumulated in a histogram. Binarizing the histogram ($= 0$, > 0) yields the nine-intersection. In a more sophisticated algorithm one would consider the dimension of the intersection sets, and reduce the overlay to the cells of the relevant dimension.

Winter presented also a data structure to store and access the cells and their skeleton. If the raster is of size $n \times m$, additional elements in a hybrid raster are $(n+1) * m$ horizontal edges, $n * (m+1)$ vertical edges, and $(n+1) * (m+1)$ nodes: the required memory space is of order 4 higher than for the raster. Another critical point of such data structures are the considerable amount of index transformations for each access.

However, if the hybrid raster is used only to represent regions – as raster does –, and no lines or points, then the additional elements of the hybrid raster become totally redundant to the cells. The skeleton can be renounced from explicit storage, applying dependency rules instead, which work locally. This paper will investigate these ideas, using a functional approach to specify semantically the rules and their application.

3 Topological Relations in a Functional Extended Raster Representation

The determination of the nine-intersection is simple in a raster representation, if the topologically completed raster is used (Section 2.3). But this does not seem practical, as the model includes not only the cells, but also the edges and the nodes; this would quadruple the storage requirement and also make computation four times longer. We will develop now a functional extension of the raster that fulfills all the conditions of a hybrid raster virtually, without explicit representation. The functions are specified in Gofer [11].

3.1 Specification of a Hybrid Raster in Natural Language

The hybrid raster representation can be computed from the regular raster representation, i.e. the necessary information is already contained in the raster, and all additional elements are redundant.

Assume an arbitrary region – without loss of generality let us confine ourselves to simple regions – given as the set of resels with value 'Region', and the background resels have the value 'Empty'. These two values are mapped to the Boolean values true and false, to allow the regular logical operations.

Cells: Cells are identical to resels. △

Vertical edges: A vertical edge belongs to the interior of the region, iff the adjacent left and right cells are labeled as 'Region'. It belongs to the exterior of the region, iff the adjacent left and right cells are labeled as 'Empty'. It belongs to the right boundary, iff the adjacent left cell is 'Region' and the right cell is 'Empty', otherwise it belongs to the left boundary (Figure 2). △

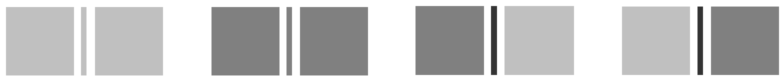

Fig. 2. Classification of edges by the two adjacent cells (bright cells are outside of the region, dark cells are elements of the region). An edge belongs (a) to the exterior or (b) to the interior, if both raster elements are homogenous, (c) and (d) to the boundary, if the values of the raster elements are different.

Horizontal edges: A horizontal edge belongs to the interior of the region, iff the adjacent upper and lower cells are labeled as 'Region'. It belongs to the exterior of the region, iff the adjacent upper and lower cells are labeled as 'Empty'. It belongs to the lower boundary, iff the adjacent upper cell is 'Region' and the lower cell is 'Empty', otherwise it belongs to the upper boundary. △

To distinguish the orientation of the boundary is not necessary in the context of this paper. But it could get importance in other tasks like line following.

Nodes: A node belongs to the interior of the region, iff all four adjacent cells are labeled as 'Region'. It belongs to the exterior of the region, iff all four adjacent cells are labeled as 'Empty'. Otherwise the node belongs to the boundary (Figure 3). △

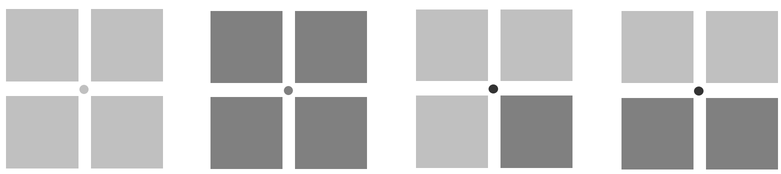

Fig. 3. Classification of nodes by the four adjacent raster elements (bright resels are outside of the region, dark resels are elements of the region). A node belongs to the exterior, if all resels are outside (a), to the interior, if all resels are element of the region (b), and to the boundary if the four resels are not homogenous; the given examples (c, d) are not complete.

To transform the rules into a formal language, an identification of each single elementsis required. We define the following index schema:

Cell index: Cells are indexed in the regular way of resels. △

Vertical edge index: Vertical edges are indexed with the same index as the cell to their left. △

Horizontal edge index: Horizontal edges are indexed with the same index as the cell above. △

Node index: Nodes are indexed with the same index as the cell left above. △

Figure 4 shows that this indexing schema is indeed complete for the Euclidean plane and gives for each element of the representation a unique index. However, any subset of the plane will miss the edges and nodes at the left and upper border by this indexing schema. For that reason it is presumed that the subsets are chosen with a border of at least one resel width ('Empty') around the represented region.

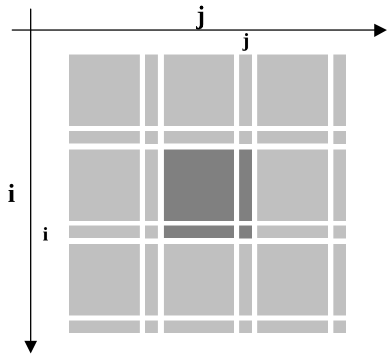

Fig. 4. Indexing schema for egdes and nodes.

3.2 Specification of a Hybrid Raster in a Functional Language

In Gofer an array can be realized as a class of `{bounds, [index := value]}`, where `bounds` are the lower and upper limit of indices, and the remainder is a list of associations between an index and a value. In the context of this paper arrays are two-dimensional and rectangular, indices are integer tuples, and the type of cells is Boolean:

```
instance Arrays (Int,Int) Bool
```

Let us extract an arbitrary 2-by-2 sub-array from a binary raster image, by applying the class method `getSubMat`:

```
get22Mat image i j = getSubMat image ((i,j),(i+1,j+1))
```

The sub-array contains the four resels (i, j), $(i + 1, j)$, $(i, j + 1)$, and $(i + 1, j + 1)$. In the following, they are referred to as patterns `cIJ`, `cEast`, `cSouth`, and `cSouthEast`, cf. Figure 5.

All the following functions map this sub-array onto a Boolean. They represent the rules of Section 3.1:

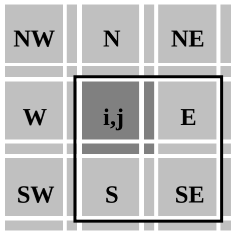

Fig. 5. The names of resels/cells in a window at cell (i, j).

```
cInterior mat22 = cIJ
cExterior mat22 = not (cInterior ar)
vInterior mat22 = cIJ && cEast
vExterior mat22 = (not cIJ) && (not cEast)
vBoundary mat22 = not (vInterior ar) && not (vExterior ar)
```

and so on for horizontal edges and for nodes. The result confirms or falsifies the rule name; e.g. if `vBoundary` returns true then the vertical edge (i, j) belongs to the boundary. While the resels are binary, the skeleton elements are ternary.

With the functions above the elements of a hybrid raster can be derived from a raster on demand at any raster position. That ability allows the construction of the hybrid raster on the fly during the overlay of two raster images. No storage of the results is specified for the functions.

3.3 Determination of Topological Relations

Testing for the intersection between boundary and interior of two simple regions determines their topological relation. In a hybrid raster, the tests must be repeated for cells, for horizontal and vertical edges, and for nodes. With regard to the limited dimension of some intersection sets, some of these tests can be neglected.

For two hybrid raster images (of the same resolution, the same orientation, and the same origin), only cells intersect with cells, only edges intersect with edges, and only nodes intersect with nodes. That is a consequence of the regular decomposition of the plane, and exceeds the usual properties of vector representations. Taking advantage from these properties, the four intersection sets of Equation 1 can be reformulated as:

```
ii_intersect a b = (&& (cInterior a) (cInterior b))
ib_intersect a b = (&& (vInterior a) (vBoundary b)) ||
                   (&& (hInterior a) (hBoundary b))
bi_intersect a b = (&& (vBoundary a) (vInterior b)) ||
                   (&& (hBoundary a) (hInterior b))
bb_intersect a b = (&& (nBoundary a) (nBoundary b))
```

Here the arguments a and b stand for two sub-arrays, one from each raster image, with the same index. That means that with this compact code at any raster position (i, j) the four intersection sets between two region interiors and boundaries are determined:

```
fourIntersectionIJ a b i j = [ii, ib, bi, bb] where
    ii = ii_intersect (get22Mat a i j) (get22Mat b i j)
    ib = ib_intersect (get22Mat a i j) (get22Mat b i j)
    bi = bi_intersect (get22Mat a i j) (get22Mat b i j)
    bb = bb_intersect (get22Mat a i j) (get22Mat b i j)
```

The remaining task is to move the 2-by-2 sub-arrays over both rasters in parallel. So the determination of the four-intersection is reduced to a convolution:

```
fourIntersection a b = ((map or).transpose)
    [ fourIntersectionIJ a b i j |
            i<-[begRow .. endRow], j<-[begCol .. endCol] ]
```

In the code a test is added to guarantee the identical image sizes. Also the patterns `begRow` and `endCol` are defined in the code, with exploitation of the outer band of 'Empty' in both images.

Let us consider the last function in more detail. Convolution yields a list of four-intersections for each raster position (right hand of the equation), which are realized as lists of four Booleans. Transposing this list of lists yields a list of four lists each containing all Booleans regarding one intersection set for the whole overlaid images. The `map` operation applies the argument – the `or` function – to all elements of the lists: we derive four Booleans for the global four intersection sets.

Extension of the procedure to the nine-intersection is straight forward.

3.4 Computational Improvements

In functional languages, the optimization is easily performed – but it is not even necessary. Languages like Gofer are 'lazy'; they evaluate functions only when needed, and only to a degree that is needed. While lazy evaluation optimizes program execution of the Gofer interpreter, the effects must be made explicit for translation to standard programming languages.

Partly the given Gofer code is already optimized: consider the limitation of evaluating intersection sets with hybrid elements of specific dimensions only. For example, `ii_intersect` evaluates only cells – no edges or nodes. That is sufficient because if the interior-interior-intersection set is not empty it must contain two-dimensional elements. – Open for optimization is the last function `fourIntersect`. The `or`, mapped to a list of Booleans, is true if at least one element is true. In principle it is sufficient to stop evaluation of each intersection set when the first true result is found.

Once optimization is done (and tested), the code can be translated into standard programming languages, like Pascal or C++.

4 Examples

Because Gofer is an executable (interpreted) language, one can run the code with some test cases. To generate such examples, first a constructor is called to deliver a raster image, initialized as 'Empty':

```
imgEmpty = binArray (-1) (-1) 2 3 False
```

Note that the bounds yield a 4-by-5 array, where the usable indices 0...1 or 2 guarantee the outer band of 'Empty' resels. – With the same constructor now two rectangular regions are created. Each region is combined with the empty image, creating the two raster images `imgA` and `imgB` (Figure 6):

```
boxA = binArray 0 0 1 0 True
boxB = binArray 0 1 0 2 True
imgA = imgEmpty // assocs boxA
imgB = imgEmpty // assocs boxB
```

More complex regions could be generated iteratively. Now we can formulate the query:

```
? fourIntersection imgA imgB
```

The result is: `[False, False, False, True]`. That means the only intersection set of Equation 1 (here in linear order) that is not empty is the intersection between the two (one-dimensional!) boundaries. The topological relationship between region A and B must be MEET therefore.

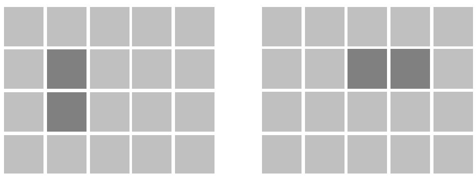

Fig. 6. The regions A (in the left raster image) and B (in the right raster image) meet along an implicit one-dimensional common boundary.

5 Conclusions

The systematic and conform extension of the topological relations, as defined by Egenhofer, from the vector representation to the raster representation can be achieved using the conceptual transformation of the raster representation into the hybrid raster, as a complete topological model. This seems not practical, but a careful examination shows that no representation for the hybrid raster representation must be constructed, and the necessary parts can be computed on the fly from a regular raster representation.

The approach to specify in a functional language yields a semantically clear piece of code that can be run with test application to demonstrate the correctness in the investigated test cases. The systematic development and the application of standard methods of program simplification and optimization leads from a conceptually simple and correct formalization to efficient operations, which can be coded in various languages. For example, a translation into C++ took only few hours including testing. Differences between Gofer specification and C++ implementation concern the conceptual change to an algorithmic language, and some adaptions to specific efficiency properties fo the target language. It is interesting to compare the codes.

In the paper effects are not investigated that originate in resolution of vector-raster conversion. We do not claim that an operation on a pair of vector regions results in the same topological relation than applied on the rasterized regions. Instead we claim in this paper that the behavior of vector and raster representation can be assimilated, by extending the raster with its skeleton. So far, the paper contributes to the merging of raster and vector operations. With the use the same conceptual operations in both representations, the difference between both can be hidden in one more instance.

It is to expect that in principle the ideas are applicable to quad-trees, too. But one has to take care of neighboring quad-tree leaves of different size. The construction of that skeleton is open to formalization. Furthermore, translation of the given specifications into standard programming languages is open for further elaboration. Only then evidence can be given for time consumption of the algorithms. We expect that the requirements are not bad, because the complexity of the problem is $O(n * m)$ with the number $n * m$ of raster elements.

References

1. Bird, R., and P. Wadler, *Introduction to Functional Programming*, Series in Computer Science, Prentice Hall International, New York, 1988.
2. Bittner, T., and A. U. Frank, *On representing geometries of geographic space*, in 8th International Symposium on Spatial Data Handling, T. K. Poiker and N. Chrisman, eds., Vancouver, 1998, International Geographical Union, pp. 111–122. 2774.
3. Dorenbeck, C., and M. J. Egenhofer, *Algebraic optimization of combined overlay operations*, in Auto-Carto 10, ACSM-ASPRS, D. M. Mark and D. White, eds., Baltimore, 1991, ACSM-ASPRS, pp. 296–312.
4. Egenhofer, M. J., *Reasoning about binary topological relations*, in Advances in Spatial Databases (SSD '91), O. Günther and H.-J. Schek, eds., Springer, 1991, pp. 143–160.
5. Egenhofer, M. J., E. Clementini, and P. di Felice, *Topological relations between regions with holes*, International Journal of Geographical Information Systems, 8 (1994), pp. 129–142.
6. Egenhofer, M. J., A. U. Frank, and J. P. Jackson, *A topological data model for spatial databases*, in Design and Implementation of Large Spatial Databases,

A. Buchmann, O. Günther, T. R. Smith, and Y.-F. Wang, eds., vol. 409 of Lecture Notes in Computer Science, Springer, New York, 1989, pp. 271–286.

7. Egenhofer, M. J., and R. D. Franzosa, *Point-set topological spatial relations*, International Journal of Geographical Information Systems, 5 (1991), pp. 161–174.
8. Egenhofer, M. J., and J. R. Herring, *A mathematical framework for the definition of topological relationships*, in 4th International Symposium on Spatial Data Handling, Zürich, 1990, International Geographical Union, pp. 803–813.
9. Egenhofer, M. J., and J. Sharma, *Topological relations between regions in* $\mathbb{R}^2$ *and* $\mathbb{Z}^2$, in Advances in Spatial Databases, D. Abel and B. Ooi, eds., vol. 692 of Lecture Notes of Computer Science, Springer, New York, 1993, pp. 316–336.
10. Frank, A. U., and W. Kuhn, *A specification language for interoperable GIS*, in Interoperating Geographic Information Systems, M. F. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, eds., Kluwer, Norwell, MA, to appear.
11. Frank, A. U., W. Kuhn, W. Hölbling, H. Schachinger, and P. Haunold, eds., *Gofer as used at GeoInfo/TU Vienna*, vol. 12 of GeoInfo Series, Dept. of Geoinformation, TU Vienna, Vienna, Austria, 1997.
12. Hernández, D., *Qualitative Representation of Spatial Knowledge*, Springer, Berlin, 1994.
13. Kong, T. Y., and A. Rosenfeld, *Digital topology: Introduction and survey*, CVGIP 48, (1989), pp. 357–393.
14. Kovalevsky, V. A., *Finite topology as applied to image analysis*, Computer Vision, Graphics, and Image Processing, 46 (1989), pp. 141–161.
15. Samet, H., *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990.
16. Winter, S., *Topological relations between discrete regions*, in Advances in Spatial Databases, M. J. Egenhofer and J. R. Herring, eds., vol. 951 of Lecture Notes in Computer Science, Springer, 1995, pp. 310–327.
17. Winter, S., *Location-based similarity measures for regions*, in ISPRS Commission IV Symposium, Stuttgart, Germany, 1998.