Control and Cybernetics

vol. **35** (2006) No. 1

Bitmap based structures for the modeling of fuzzy entities

by

Jörg Verstraete, Guy De Tré, Axel Hallez

Department of Telecommunications and Information Processing
Ghent University
Sint Pietersnieuwstraat 41, 9000 Ghent, Belgium
e-mail: jorg.verstraete, guy.detre, axel.hallez@telin.ugent.be

Abstract: Bitmap models are a known technique to model field based geographic information. Commonly, geographic information is modelled in a crisp sense, even though in reality it most likely is an approximation. In this article, we present the use of bitmap based structures to model imprecise or uncertain locations and ditto regions; these structures should be considered to be extensions of respectively a point and a polygon. The imprecission or uncertainty is modelled using fuzzy set theory. Apart from presenting the structures, appropriate operators are defined and explained.

Keywords: imprecise GIS, uncertain GIS, fuzzy data modelling, fuzzy bitmaps.

1. Introduction

Traditional geographic information systems use basic geometry elements (i.e. points, lines, arcs, polygons) to model features (Rigaux, Scholl, Voisard, 2002; Shekhar, Chawla, 2003). A feature, in essence an object on the map, can be a point (i.e. precise location), a line or a *polyline*[1] (i.e. to indicate roads or rivers) or a polygon (i.e. a region, a lake). Using these basic geometry elements, only crisp data can be modeled: a feature is set at one given location; a region has a fixed surface area and ditto boundary, associated data (numerical or text data associated with a position) are crisp. In reality, it turns out that lots of data are inherently prone to imprecision or uncertainty: the exact location of a feature can be imprecisely known (i.e. the whereabouts of a person or the location of a lightning strike in an open field), an area can have a non-crisp border (i.e. in soil compositions: the border between a sandy soil and clay),

[1] A polyline is a set of connected lines: the endpoint of one line is the starting point of another line.

associated data can be vague or uncertain (i.e. predictions of temperatures, or population numbers), etc. Traditional systems can not take this inherent vagueness (caused by imprecision, uncertainty or a combination of these) into account, resulting in the fact that the traditional model is an approximation of reality without any indication regarding the quality of the approximation, nor how much it deviates from reality. Modelling just this would enrich the model a great deal, (see Morris, 2001), yielding a better representation of reality which in turn would provide more realistic analysis and predictions.

In this article, an approach for the modelling of vaguely defined regions (this can be seen as an extension of the traditional concept of polygon) and positions (this can be seen as an extension of the traditional concept of a point) is presented. It is important to realize that - while objects (positions and regions) will be modelled - the *field based* concept of a bitmap is used. However, as the bitmaps are used in a different sense than is more common, this will require some specific operators: the term *fuzzy bitmap* is used to indicate bitmaps in this usage.

For the modelling of imprecision and vagueness, fuzzy set (Zadeh, 1975; Prade, 1982; Zimmerman, 1999; Dubois, Prade, 2000, 2001) theory is used.

2. Definition of a fuzzy bitmap

A fuzzy bitmap is in essence an extension of a regular, crisp bitmap. Similar to a regular bitmap, a fuzzy bitmap is considered to be limited to a certain, crisp region (the region of interest). In order to formally define a fuzzy bitmap, first the concepts of *cell* and *grid* will be defined.

With the understanding that X is the universe of all the locations (points) considered in the GIS, a subset $c \subseteq X$ is called a cell if it is convex, i.e.:

$$\forall p_1, p_2 \in c,\ \exists p_3 \in c : \frac{\overrightarrow{p_1} + \overrightarrow{p_2}}{2} = \overrightarrow{p_3}. \tag{1}$$

The cell is the smallest unit known to the bitmap; for some operators the center point of a cell is used as the reference point for this cell. This point will be denoted as p^c.

A *grid* — in this context — partitions the region of interest R in a finite collection of disjoint cells

$$G = \{c \subseteq X | \forall c_1, c_2 \in G : c_1 \cap c_2 = \emptyset; \bigcup_{c_i \in G} = R\}. \tag{2}$$

In Verstraete et al. (2005), the bitmap was considered a global structure; now it is limited to a region of interest. This difference is resembled in this altered definition of a grid. In general, all cells have similar shapes and sizes, although the fuzzy bitmaps are not limited to this: in the examples here cells will be rectangular, but the length and width proportions of cells can differ. Each grid has a fixed number of horizontal and vertical cells. Other shapes of cells, i.e. hexagonal, are possible but not considered here.

Similarly to traditional bitmaps, a value will be associated with every cell of the bitmap. In a fuzzy bitmap, these values are limited to the range $[0, 1]$ as they will represent membership grades.[2] The membership function associates every cell of a grid with its membership grade for a given bitmap B:

$$\mu_B : G \rightarrow [0,1] \tag{3}$$

$$c \mapsto \mu_B(c). \tag{4}$$

The definition of a fuzzy bitmap $\tilde{B}$ using grid G and membership function μ then is:

$$\tilde{B} = \{(c_j, \mu_B(c_j)) | c_j \in G\}. \tag{5}$$

This definition differs from the one in Verstraete et al. (2005) in that the cells are no longer numbered using two indices (coordinates), but only using one index. The reason for this is twofold: first to accommodate the definition for both regular and irregular grids; second, this numbering also matches the numbering of cells in Esri MapObjects, which is used for a prototype implementation. The downside to this numbering is that given a cell, its relative position to other cells of the same bitmap is not immediately known. Several bitmaps can be defined using the same grid. This means that they cover the same region of interest and that their cells are exactly the same size; the associated values of the cells can differ, though.

3. Using fuzzy bitmaps as fuzzy regions

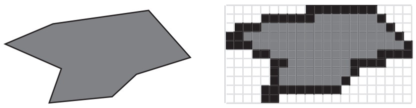
3.1. Using bitmaps to represent crisp regions

A region in a GIS is commonly represented by means of a polygon. It is possible, however, to use a bitmap to represent geometric figures, using algorithms like those used to display vector graphics on a computer screen. In Fig. 1a, a polygon is shown; Figs. 1b and 1c show the same polygon in a bitmap representation. In Fig. 1b the grid is also shown, it is more coarse than in Fig. 1c (where the grid has been omitted for the readability of the image). It is obvious that a more refined grid will lead to a better approximation of the polygon.

3.2. Using bitmaps to represent fuzzy regions

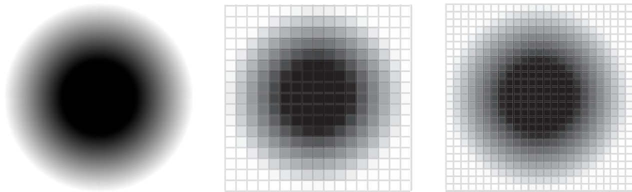
In this section, the concept of fuzzy regions is introduced. Conceptually, a fuzzy region is a region with an imprecisely or uncertainly defined boundary.

[2] The membership grade 0 is included in a bitmap as this facilitates the implementation by allowing all bitmaps to bounded by a rectangular region of interest (the added cells then are assigned membership grade 0.) For any operation, they can simply be ignored; the value 0 then serves as a dummy value to identify these cells.



(a) (b) (c)

Figure 1. Use of a bitmap to approximate a crisp region.



(a) (b) (c)

Figure 2. Use of a bitmap to approximate a fuzzy region.

Various representations exist, most of which use some variant of the egg-yolk model (Cohn and Gotts, 1994; Gotts and Cohn, 1995). Some have extended the egg-yolk model to provide a model of the boundary itself, i.e. Beaubouef, Petry (2001), Hallez, Verstraete (1995), Clementini (2004). The main difference between the traditional egg-yolk model and our approach is that the egg-yolk model does not provide a model for the broad boundary itself; in our approach each point of the fuzzy region - which includes the broad boundary - is assigned a value[3] to indicate to what extent it belongs to the region. In this bitmap approach, a fuzzy region is represented as a fuzzy bitmap $\tilde{B}$, where the membership grade associated with every cell is the extent to which this cell belongs to the region. This interpretation, in which all cells belong to the region (but some only to a given extent) is an example of what is called in fuzzy set theory a *veristic* interpretation (Dubois, Prade, 1997).

In Fig. 2a, a simple fuzzy region is shown. For representation purposes, greyscales are used: black equals membership grade 1, the lower the membership grade of a point (or in the case of the bitmap: a cell) is, the lighter its shade of grey. Figs. 2a and 2b show a representation of this fuzzy region in the bitmap approach. The grid used in Fig. 2c is more refined than the grid used in Fig. 2b, which - as already was shown in the crisp case - yields a more accurate model.

A fuzzy region can be used for various purposes: soil composition (i.e. indicating clay ground), population densities, etc. A special case occurs when the fuzzy region is interpreted as possible locations for a point, in which case it will be referred to as a fuzzy point, as explained in the next Section.

It is important to emphasize that while the traditional concept of a bitmap

[3]The bitmap approach does not work on a point basis, but on a cell basis; different points are grouped in cells, and this value is assigned to cells.

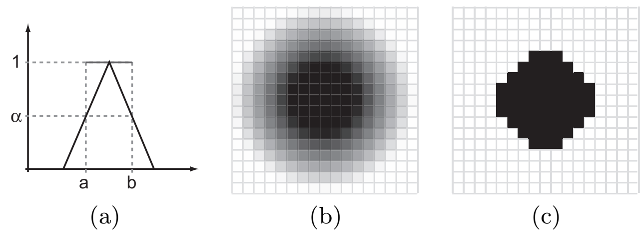


Figure 3. Alphacuts for bitmaps.

is a field based model, our approach uses basically the same concept on which a feature based model supporting imprecission and uncertainty is defined. Consequently, for the remainder of the paper, a fuzzy bitmap will either be a fuzzy region (Section 3.3) or a fuzzy point (Section 4).

While for practical uses the bitmap model suffers from the fact that it is a discrete model, it can still serve some applications. Being a discrete model, the bitmap model is particularly interesting for theoretical purposes, as this facilitates the definition of various operators. Ongoing work is also aimed at development of a similar, more accurate model.

3.3. Operations on single fuzzy bitmap-regions

3.3.1. α-cut

When working with fuzzy structures, at some point there will be the need to *defuzzify* information, which implies there must be means to omit everything fuzzy. This can be needed for instance to display the results, but also to make it possible for a fuzzy model to be exported to a system that has no support for fuzzy models, or to a system that supports another model for fuzzy geographic information. As many extensions of geographic operators presented here make use of α-cuts, they are considered first.

Traditionally in fuzzy set theory, the α-cut operator is used for defuzzification: the α-cut of a fuzzy set returns all the elements which have a membership grade greater than a given threshold. Elements whose membership grade is not greater than this threshold are not in the result set. Fig. 3a is an illustration of this.

In the bitmap model, the α-cut takes a fuzzy bitmap as argument (Fig. 3b) and results in a new fuzzy bitmap, as illustrated on Fig. 3c. The cells of this new fuzzy bitmap will only have associated values 0 or 1. The resulting bitmap will share the same grid as the bitmap used as argument:

$$\tilde{G}_{result} = G_{orig}. \tag{6}$$

In fuzzy set theory, a difference is made between a *strong α-cut* and *weak α-cut*; this difference is also reflected in our model. The *strong α-cut* of a fuzzy set returns the elements with a membership grade strictly greater than a given threshold:

$$\tilde{B}_{\overline{\alpha}} = \{(c_j, 1) | \mu_{B_{orig}}(c_j) > \alpha, c_j \in G\}. \tag{7}$$

A special case of a strong α-cut is the *support*; this is the strong alpha-cut with threshold 0. This is an important alpha cut, as it results in all the elements that belong to some extent to the fuzzy set.

$$\tilde{B}_{\overline{\alpha_0}} = \{(c_j, 1) | \mu_{B_{orig}}(c_j) > 0, c_j \in G\}. \tag{8}$$

Analogous to the strong α-cut, the *weak α-cut* of a fuzzy set returns the elements with a membership grade greater than or equal to a given threshold:

$$\tilde{B}_{\alpha} = \{(c_j, 1) | \mu_{B_{orig}}(c_j) \geq \alpha, c_j \in G\}. \tag{9}$$

Similarly to the strong α-cut, the weak α-cut has a special case, now for a threshold equalling 1. This α-cut is called the *kernel*, and returns all the elements that fully belong (membership grade 1) to the given fuzzy set:

$$\tilde{B}_{\alpha_1} = \{(c_j, 1) | \mu_{B_{orig}}(c_j) \geq 1, c_j \in G\}. \tag{10}$$

3.3.2. Surface area

For the calculation of the surface area of a fuzzy region, there are two possible interpretations. The first is when the surface area is interpreted as a measurement for the area. For a fuzzy region, this will mean its surface area will be a fuzzy number. The second interpretation is when the surface area is considered to be an expression of fuzzy cardinality (Klir, Yuan, 1995); in this case the surface area of a fuzzy region will be the cardinality of the fuzzy set and will thus be a crisp number. Both interpretations are considered below.

The calculation of the fuzzy surface area $\tilde{S}$ of a fuzzy bitmap $\tilde{B}$ makes use of the previously defined α-cut. Conceptually, the surface area of each weak α-cut will be used to determine the fuzzy number that represents the surface area. Similar to the calculation of the distance, first the available α-cuts are considered. In practice, only the α-cuts at membership grades present in $\tilde{B}$ will need to be considered:

$$0 < \alpha_0 < \alpha_1 < ... < \alpha_n \leq 1 \tag{11}$$

where $\exists c \in \tilde{B} : \mu_{\tilde{B}_1}(c) = \alpha$. Based on each of these α-values, S_α can be defined:

$$S_\alpha = \sum_{\mu_{\tilde{B}(c_j)} > \alpha} S(c_j), \forall c_j \in B. \tag{12}$$

Using these S_α, the fuzzy surface area $\tilde{S_\alpha}$

$$\mu_{\tilde{S}(\tilde{B})}(x) = \begin{cases} \alpha_i & \text{if} \quad \mathrm{S}_{\alpha_\mathrm{i}} \leq \mathrm{x} < \mathrm{S}_{\alpha_{\mathrm{i}+1}} \\ \alpha_n & \text{if} \quad \mathrm{x} = \mathrm{S}_{\alpha_\mathrm{n}}. \end{cases} \tag{13}$$

As mentioned before, the surface area of a region can also be interpreted as a cardinality (in a sense, it counts the number of points in that region). For a fuzzy region, this interpretation for the surface calculation is equivalent to the notion of fuzzy cardinality (Klir, Yuan, 1995). In this concept, the surface area of each cell is considered and its associated membership grade will be used to determine just how much this area contributes to the total (cardinality of the) area:

$$card(\tilde{B}) = \sum_{c \in \tilde{B}} (S(c) \times \mu(c)). \tag{14}$$

3.3.3. Minimum bounding rectangle

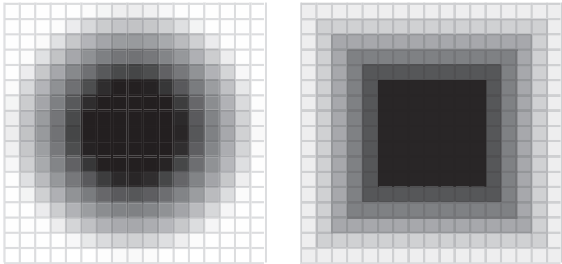
In traditional GIS systems, a minimum bounding rectangle of a polygon is the smallest rectangle that can contain the polygon, and the sides of which are parallel to the axes used (Rigaux, Scholl, Voisard, 2002). This concept can be used for a number of purposes, ranging from determining the relative position of two features to optimizing operators (i.e. if the MBRs of two regions do not overlap, the regions do not overlap). For a fuzzy region, two variants of the concept of an MBR are considered. The first is a fuzzy minimum bounding rectangle, which results in a fuzzy defined rectangle (i.e. another bitmap structure); the second requires an alpha level, and results in a crisp rectangle bounding this alpha level.

The concept of the fuzzy minimum bounding rectangle as introduced here should not be confused with the fuzzy minimum bounding rectangle defined in Somodevilla and Petry (2004), where the authors define both the minimum bounding rectangle and the inscribed rectangle of a fuzzy region, along with a number of intermediate rectangles, in order to approximate this region.

A fuzzy MBR will yield a bitmap with the same grid as the original bitmap. The fuzzy MBR of a bitmap will be a new bitmap whose every α-cut is rectangular. The fuzzy MBR bitmap is defined such that these rectangular α-cuts are MBRs for the same α-cuts of the original bitmap. This is illustrated on Fig. 4: Fig. 4a shows the original bitmap, Fig. 4b shows its fuzzy MBR.

As a bitmap holds a finite number of cells, it also holds a finite number of membership grades. Consequently, only these grades that are present in the bitmap should be considered as α-cuts to define the fuzzy MBR.

Consider all these α-cuts. With each α-cut, a bitmap-MBR can be defined: a rectangle made of cells such that all cells belonging to this α-cut are inside the rectangle and no smaller rectangle can be defined. Such a rectangle can be considered for each α-level, and each rectangle can be considered as a bitmap. The union of all these (overlapping) bitmaps - as explained in Section 3.5.3 - yields a new bitmap. The construction of this bitmap-MBR is explained below in pseudo-code.



(a) (b)

Figure 4. Concept of the fuzzy MBR.

```
Fuzzy_Bitmap Fuzzy_MBR (Fuzzy_Bitmap B)
result_x, temp_x: fuzzy_bitmaps, same grid as B, all grades=0
BEGIN
  determine available alpha levels in B

  for each alpha level x
    determine B_x

    find the cells with grade=1 in B_x that are closest to the
      left/right/top/bottom side of the grid
    use these cells to define a rectangle in temp_x:
      leftmost cell determines lefthand side of the rectangle
      rightmost cell determines righthand side of the rectangle
      topmost cell determines tophand side of the rectangle
      bottommost cell determines bottom side of the rectangle

      cells inside this rectangle are assigned membership grade 1
      cells outside this rectangle are assigned membership grade 0

    for each cell with grade=1 in temp_x
      if the same cell in result has a value < x
        assign this cell the value x in result
      end for
  end for
  return result_x
END
```

This new bitmap has the property that its α-cuts are MBRs for the same α levels in the original bitmap, this new bitmap is considered as the fuzzy MBR. Note that membership grades in the fuzzy MBR will always be concentric: higher grades towards the middle, lower grades towards the edge of the fuzzy bitmap.

The fuzzy MBR has the disadvantage that it still is a bitmap-structure, making it impossible for existing systems to use this information without modifications. The crisp MBR of a bitmap is a polygon (a rectangle), just like any MBR in traditional systems. In addition to a bitmap, the calculation of a crisp MBR also requires an α level: this level determines the cells of the bitmap around which the MBR is considered.

The crisp MBR can be calculated as follows: first, the fuzzy MBR is determined. Next, the α-cut at the given level is considered. This will yield a bitmap with cells having an associated membership grade 1 and cells with an associated grade 0. The outline of the cells with an associated grade 1 can now be represented as a polygon (by construction it will be a rectangle), resulting in a traditional MBR.

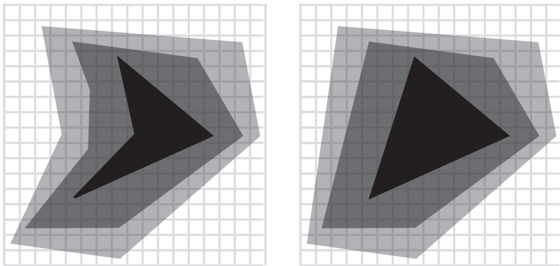
3.3.4. Convex hull

The convex hull of a polygon (Rigaux, Scholl, Voisard, 2002) is an interesting operator in traditional GIS systems. It is commonly used to optimize other operators and tests, i.e. if the convex hulls of two polygons do not intersect, the polygons themselves do not intersect. It can also be used for indexing. Even in this usage, fuzzification makes sense: the same fuzzy region can have index entries for different α-levels.

Traditionally, the convex hull of a polygon results in a new polygon; for fuzzy regions, the convex hull of a fuzzy region will result in a new fuzzy region. The approach is quite similar to the calculation of the fuzzy MBR: for every α-cut, the convex hull is considered. By recombining these results using the union operator, a new bitmap containing the fuzzy convex hull is obtained. Fig. 5 shows a simplified example (the cells are not considered) to illustrate the concept: Fig. 5a shows a fuzzy region, its fuzzy convex hull is shown on Fig. 5b.

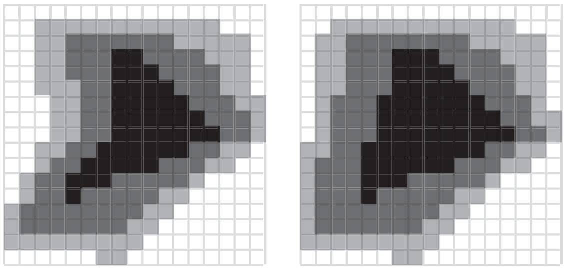
Similar to the calculation of the fuzzy MBR, only the α-levels at membership grades that occur in $\tilde{B}$ need to be considered.

```
Fuzzy_Bitmap Fuzzy_Convex_Hull (Fuzzy_Bitmap B)
result_x, temp_x: fuzzy_bitmaps, same grid as B, all grades=0
BEGIN
  determine available alpha levels in B

  for each alpha level x
    determine B_x

    consider the centerpoints of cells with grade=1 in B_x that
      neighbour cells with grade=0 or
      neighbour the edge of the bitmap

generate convex hull of polygon defined by these centerpoints
```



(a) (b)

Figure 5. Simplified illustration of the concept of the fuzzy convex hull.



(a) (b)

Figure 6. Example of a fuzzy convex hull of an extended bitmap.

```
rasterize the polygon (using the current grid as raster)
  if a cell belongs to the edge or the inside of the polygon
    assign it grade=1 in temp_x

    for each cell with grade=1 in temp_x
      if the same cell in result has a value < x
        assign this cell the value x in result
      end for
  end for
  return result_x
END
```

In the algorithm, a rasterization-method is required. These methods are common in the realm of computer graphics, for a description of different rasterization-techniques we refer to Foley and Feiner (1996), Angel (2003).

On Fig. 6a fuzzy bitmap is considered (it is the rasterized example of Fig. 5. The fuzzy bitmap is shown on Fig. 6b. This result might not appear to be convex, but a bitmap representation is limited in that it can only consider cells

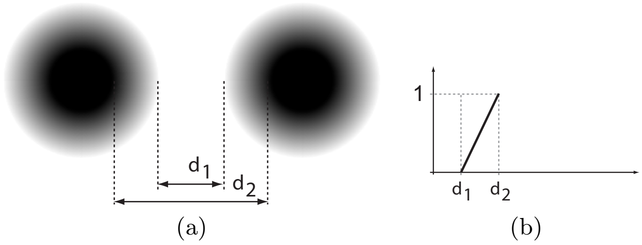


Figure 7. Concept of distance between fuzzy regions.

as its smallest unit. The bitmap usually is an approximation of a polygon, the convex hull of a bitmap will also be an approximated polygon. The fuzzy bitmap as constructed above has the property that at every α-level it holds the convex hull for the original bitmap (at that same α-level).

3.4. Operations on multiple fuzzy bitmap-regions with non-spatial result

3.4.1. Distance between fuzzy bitmaps

The geographic operator that will be considered here is the distance operator. Only the distance between two fuzzy regions is explained; the distance between a fuzzy and a crisp region is analogous.

In the crisp case, the distance between two regions is defined as the shortest distance of all possible distances between these two regions; this definition is the basis for defining the distance between two fuzzy regions:

$$d(R_1, R_2) = min(d(p_1, p_2), \forall p_1 \in R_1 \wedge \forall p_2 \in R_2). \tag{15}$$

When dealing with fuzzy regions, it stands to reason that the result will be a fuzzy number: if the regions are imprecisely defined, so must the distance between them. This is illustrated in Fig. 7

In order to extend the distance operator, first all the membership grades of both arguments must be considered:

$$0 < \alpha_0 < \alpha_1 < ... < \alpha_n \leq 1 \tag{16}$$

where $\forall, \alpha_i \exists c \in \tilde{B}_1 \cup \tilde{B}_2 : \mu_{\tilde{B}_1}(c) = \alpha_i \vee \mu_{\tilde{B}_2}(c) = \alpha_i$. Along with each of these α-values, l_α can be defined; l_α is the shortest distance between the α-levels of the bitmaps:

$$l_\alpha = min\left(d(p_1^c, p_2^c), \forall p_1^c \in \tilde{B}_{1\alpha} \wedge p_2^c \in \tilde{B}_{2\alpha}\right). \tag{17}$$

The distance is considered between centerpoints[4] of all cells that belong to this

[4]Depending on the accuracy of the bitmaps - this will depend on the application - an alternate definition could use all the points $p_1 \in \tilde{B}_{1\alpha}$ and $p_2 \in \tilde{B}_{2\alpha}$ of the each cell instead of only the center points.

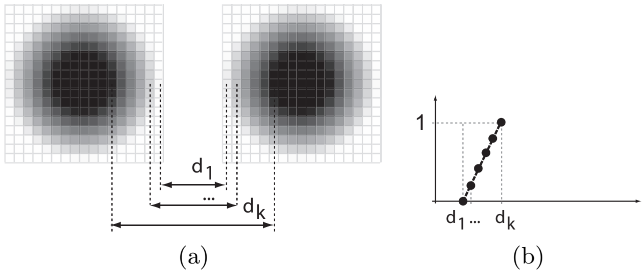


Figure 8. Example of distance between two extended bitmaps representing fuzzy regions.

α-level in each of the bitmaps; l_α is defined for all α levels occuring in both bitmaps. The distance l_{α_0} is the shortest distance that occurs; the distance l_{α_n} is the longest.

The distance between two fuzzy regions is then defined using the l_α values as:

$$\mu_{\tilde{d}(\tilde{B_1},\tilde{B_2})}(x) = \begin{cases} \alpha_i & \text{if} \quad \mathrm{x} \in [\mathrm{l}_{\alpha_\mathrm{i}}, \mathrm{l}_{\alpha_{\mathrm{i+1}}}[\\ \alpha_n & \text{if} \quad \mathrm{x} = \mathrm{l}_{\alpha_\mathrm{n}} \end{cases} \tag{18}$$

This operator is a straightforward application of the Zadeh‘s extension principle, applied to the distance as defined between regions. To illustrate the operator, consider the bitmaps B_1 and B_2 as shown in Fig. 8a. The distance between the fuzzy regions represented by these bitmaps is shown in Fig. 8b. As the distance is considered at all available α-levels, the fuzzy number appears in steps. As the extended bitmaps are an approximation for the regions, this stepped distance can be considered to be an approximation as well. A nice representation of this fuzzy number is obtained by considering its convex hull, as illustrated by the thick line in Fig. 8b.

3.5. Operations on fuzzy bitmap-regions with spatial result

3.5.1. The new grid

Traditionally in GIS, different data can be combined in what is called an *overlay.* Overlays are essential to combine different types of data, or data from different sources.

When *overlaying* multiple bitmaps, it is important to note that this usually includes a change in the region of interest. Consequently, before any operation can be considered, the new region of interest must be determined. This new region of interest will encompass the regions of interest of its arguments. Next, the arguments must also be considered within this new region of interest, which implies extending their grid to match the region.

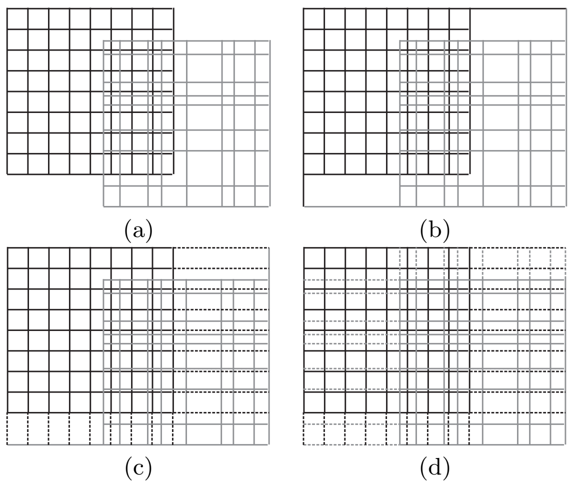


Figure 9. Constructing the grid that combines two existing grids.

In the first paragraph, defining the new region of interest is considered, as is adapting the arguments of the operators to match this new region. In the subsequent paragraphs a number of set-operations are defined.

Consider two grids as shown on Fig. 9a. The first step in defining the grid is determining the new region of interest. This region of interest is basically the union of the regions as considered by the bitmap arguments. However, as a bitmap is considered to have a rectangular outline, the region is extended in such a way that its outline be a rectangle, see Fig. 9b.

In a second step, the region of interest is partioned by the grid of the first bitmap, but with its gridlines lenghtenend beyond its own region of interest to the outline of the newly defined region of interest, thus possibly dividing cells of the other grid. The lengthened gridlines are drawn using dashed lines on Fig. 9c.

In consecutive steps, the grids of other bitmap arguments are used to partition the new region of interest (and its cells, if needed) even further. Essentially, the cells of one grid can be divided by the other grid into smaller cells. This can be seen on Fig. 9d.

The result of this construction is that every cell that was present in one of the arguments is present in the new region of interest, either as a whole, or partitioned in a number of smaller cells. Now, the original grids are discarded, and every bitmap that is an argument is now using these grids for its cell definitions (cell coordinates used below are relative to this grid), as illustrated on Fig. 10. While this action potentially changes the resolution of a bitmap, its overall appearance is not altered by this: cells either inherit their membership grade from the original bitmap, or are assigned 0 if they cover regions not covered by the original bitmap.

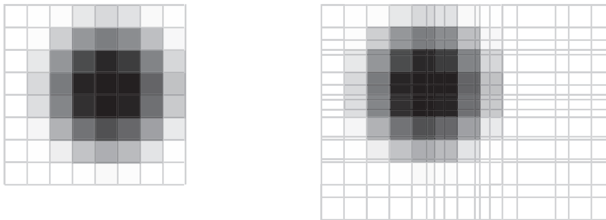


Figure 10. Mapping the bitmap from its original grid to the newly constructed grid.

3.5.2. Intersection

One way of combining data of multiple bitmaps is by considering their intersection. If the fuzzy bitmaps $\tilde{B}_1$ and B_2 model features A and B respectively, the intersection of both bitmaps will model the regions where both features A and B are present.

The intersection is performed by a T-norm operator, i.e. the minimum. As the operator is applied on a per cell basis, any T-norm can be used:

$$\mu_{\tilde{B}_3}(c_3(n,m)) = T(\mu_{\tilde{B}_1}(c_1(n,m), \mu_{\tilde{B}_2}(c_2(n,m))).$$

3.5.3. Union

The union of two bitmaps can be used to yield the regions where one of two features (each modeled by its own bitmap) occurs.

This operator is performed by a T-conorm (i.e. maximum), but again, as the operator is applied on a per cell basis, any T-conorm can be used:

$$\mu_{\tilde{B}_3}(c_3(n,m)) = S(\mu_{\tilde{B}_1}(c_1(n,m), \mu_{\tilde{B}_2}(c_2(n,m))).$$

Other operators, such as difference, are completely analogous.

4. Fuzzy bitmaps for fuzzy points

In the previous section, a bitmap was used to model a fuzzy region, where all points of the bitmap were considered to belong completely or to some extent to the region. In this section, the modelling fuzzy points is considered. A fuzzy point is in essence an extension of a point: the representation of a single position, but with uncertainty or imprecision regarding that position. In practical uses, a fuzzy point can be used to model a number of things: the estimated position of a person/object on a map derived from limited knowledge (i.e. close to a church tower, a bridge and a river; or position after a given time interval after the last know gps coordinates), matching different sources of information (i.e. identify the same crossing on a map and an aerial photograph).

By considering a fuzzy region as a model for *all the possible locations of a point*, the model for a fuzzy region can be used as a basis to represent a fuzzy point. This fuzzy region is represented as a fuzzy bitmap $\tilde{B}$, where the membership grade associated with every cell is the extent to which this cell is a possible location for the fuzzy point. The main difference is the interpretation of the bitmap: as a point is being modelled, only one location is valid at all times. Consequently, the interpretation of the fuzzy bitmap is possibilistic (a fuzzy region has a veristic interpretation: all points are valid but to a different extent). This difference in interpretation influences some operators (such as distance), whereas other operators either remain the same (perhaps with a difference in interpretation), or might even lose their meaning.

4.1. Operations on single fuzzy bitmaps (points)

The operations on single fuzzy bitmaps as described in the previous section are the same when considering fuzzy points. The various α-cuts can be useful when working with fuzzy points, the fuzzy MBR and fuzzy convex hull can be used to work on the *possible locations for a point*. Even the fuzzy surface can be used, to determine over what area the point is located.

Of course, in all these operations, one must consider the difference in interpretation: when modelling a fuzzy point, only one point (cell) is considered at a time (with fuzzy regions, all cells were considered at the same time).

4.2. GIS-operations on multiple fuzzy bitmaps (points)

In this section, traditional operators on points will be adapted to work with the concept of fuzzy points. For the sake of argumentation, consider two fuzzy bitmaps $\tilde{B_1}$, respectively $\tilde{B_2}$, used to represent the fuzzy points $\tilde{p_1}$ and $\tilde{p_2}$. The notation p will be used to indicate traditional crisp points.

Most operators are similar to the operators defined above. Only the operators that differ from the operators in the fuzzy-region section will be considered.

4.2.1. Distance between fuzzy points

The Euclidean distance d between two crisp points is defined as

$$d(p_1(x_1, y_1), p_2(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{19}$$

The distance $\tilde{d}$ between two fuzzy positions $\tilde{p_1}$, $\tilde{p_2}$ will be a fuzzy number, which can be defined using Zadeh's Extension Principle (Zadeh, 1975):

$$\mu_{\tilde{d}(\tilde{p_1},\tilde{p_2})}(x) = \sup_{\substack{p_1, p_2 \in X \\ x = d(p_1, p_2)}} min(\mu_{\tilde{p_1}}(p_1), \mu_{\tilde{p_2}}(p_2)). \tag{20}$$

This concept can be seen in Fig. 11.

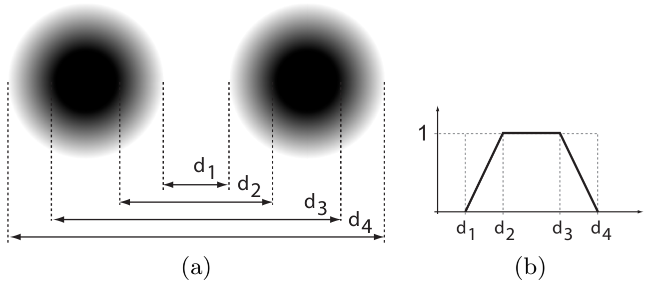


Figure 11. Concept of distance between fuzzy points.

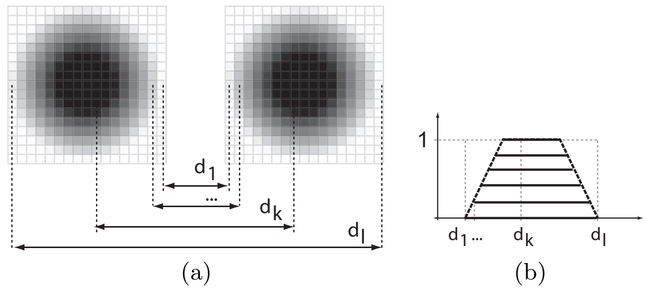


Figure 12. Example of distance between extended bitmaps representing fuzzy points.

In the case of the bitmap, the definition must work on a per cell basis. Consequently, this definition becomes

$$\mu_{\tilde{d}(\tilde{B_1},\tilde{B_2})}(x) = \sup_{\substack{c_1 \in \tilde{B_1}, c_2 \in \tilde{B_2} \\ x = d(c_1,c_2)}} min(\mu_{\tilde{B_1}}(c_1), \mu_{\tilde{B_2}}(c_2)) \tag{21}$$

This concept can be seen in Fig. 12. Similarly as the distance between regions, the obtained number can be approximated by its convex hull, thus resulting in the thick line in Fig. 12b.

Other distance measures can be defined in an analogous way.

4.2.2. Set-operations on multiple fuzzy bitmaps (points)

The set-operations as described in the section concerning fuzzy regions can be used for fuzzy bitmaps that represent fuzzy locations. Set operations in the context of fuzzy points are necessary when there is different data concerning the location of a point. For example: it can be known that a point is *near a river* and *close to a water tower*. The intersection operator will provide a means of combining these two pieces of information. To do this, first a bitmap can be constructed with possible locations *near a river*, then a second bitmap can hold possible locations that are *close to a water tower*. The intersection of these bitmaps yields the possible locations that are both close to a river and near a water tower.

5. Conclusion

In this paper, bitmap models - commonly a field based model - were extended for the modelling of fuzzy regions and fuzzy points. The presented models should be seen as extensions of polygons and points, respectively. In addition to the representation of the data, operators commonly found in GIS systems were extended to these bitmap models. The surface, bounding rectangle and convex hull of a fuzzy bitmap can be calculated, as can the distance between, and the intersection and union of two fuzzy bitmaps be determined. Additional operators that allow the defuzzification (using α-cuts) of a fuzzy bitmap increase the usability in traditional systems.

The theoretical definitions and algorithms have been implemented in a working prototype, which illustrates their feasibility. Further research is aimed at optimizing the operators and adding more operators, i.e. for determining relative positions between regions and/or points. The bitmap model is an easy model to reason upon, but it still is a discrete model. Work on a more accurate representation for fuzzy points and regions based on triangulated irregular networks is also ongoing.

References

Angel, E.S. (2003) *Interactive Computer Graphics: A Top-Down Approach with OpenGL*. Addison-Wesley.

Beaubouef, T. and Petry, F.(2001) Vagueness in Spatial Data: Rough Set and Egg-Yolk Approaches. *Proc. IEA/AIE 2001 Conf.*, Budapest, Hungary, 2001. *Eng. of Intelligent Systems: Lecture Notes in AI* **2070**. Springer-Verlag, 367–373.

Clementini, E. (2004) Modeling Spatial Objects Affected by Uncertainty. In: R. De Caluwe, De Tré G., Bordogna G., eds., *Spatio-Temporal Databases - Flexible Querying and Reasoning*. Springer-Verlag, 211–236.

Cohn, A.G. and Gotts, N.M. (1994) Spatial regions with undetermined boundaries. *Proceedings of the Second ACM Workshop on Advances in Geographic Information Systems*, 52–59.

Dubois, D. and Prade, H. (1997) The three semantics of fuzzy sets. *Fuzzy Sets and Systems* **90**, 141-150.

Dubois, D. and Prade, H. (2000) *Fundamentals of Fuzzy Sets*. Kluwer Academic Publishers.

Dubois, D. and Prade, H. (2001) Possibility theory, probability theory and multiple-valued logics: A clarification. *Annals of Mathematics and Arti-*

ficial Intelligence **32**, 35–66.

Foley, van Dam, Feiner, H. (1996) *Computer Graphics.* Addison-Wesley.

Gotts, N.M. and Cohn, A.G. (1995) A mereological approach to representing spatial vagueness. *Working Papers, Ninth International Workshop on Qualitative Reasoning*, 246–255.

Hallez, A., Verstraete, J., De Tré, G. and De Caluwe, R. (2002) Contourline Based Modelling of Vague Regions. *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU 2002*, July 1-5, Annecy, France. EISA, Université de Savoie, 1721–1726.

Klir, G.J. and Yuan, B. (1995) *Fuzzy Sets and Fuzzy Logic: Theory and applications.* Prentice Hall, New Jersey.

Morris, A. (2001) Why Spatial Databases Need Fuzziness. *Proceedings of IFSA/NAFIPS 2001*, Vancouver, Canada (CD), 2446–2451.

Prade, H. (1982) Possibility sets, fuzzy sets and their relation to Lukasiewicz logic. *Proc. 12th Int. Symp. on Multiple-Valued Logic*, 223–227.

Rigaux, P., Scholl, M. and Voisard, A. (2002) *Spatial Databases with Applications to GIS.* Morgan Kaufman Publishers.

Shekhar, S. and Chawla, S. (2003) *Spatial Databases: A Tour.* Pearson Education Inc.

Somodevilla, M.J. and Petry, F.E. (2004) Fuzzy Minimum Bounding Rectangles, In: R. De Caluwe, G. De Tré, G. Bordogna, eds., *Spatio-Temporal Databases - Flexible Querying and Reasoning.* Springer-Verlag, 237–263.

Verstraete, J., De Tré, G., De Caluwe, R. and Hallez, A. (2005) Field Based Methods for the Modeling of Fuzzy Spatial Data. In: F. Petry, V. Robinson, M. Cobb, eds., *Fuzzy Modeling with Spatial Information for Geographic Problems.* Springer-Verlag, 41–69.

Zadeh, L. (1975) The concept of a linguistic variable and its application to approximate reasoning I, II, III. *Information Sciences* **8**, 199–251, 301–357, **9**, 43–80.

Zimmerman, H.J. (1999) *Practical Applications of Fuzzy Technologies.* Kluwer Academic Publishers.