Computing Polygon Similarity from Raster Signatures

Leo Antunes[1], Leonardo Guerreiro Azevedo[1,2]

[1] Graduate Program of Infomation Systems (PPGI)
Applied Informatics Department (DIA)
Federal University of State of Rio de Janeiro (UNIRIO)
Av. Pasteur, 458, Urca, 22290-240, Rio de Janeiro, Brazil

[2] Research and Practice Group in Information Technology (NP2Tec)
{leo.antunes,azevedo}@uniriotec.br

Abstract. *Computing similarity of spatial objects is not a trivial task. It considers complex algorithms, which have high cost to compute. This work proposes a simple algorithm to compute similarity between polygons through their Four-Color Raster Signatures (4CRS) based on Jaccard index. The algorithm was implemented in SECONDO, an extensible DBMS platform. Experimental tests were conduct in order to evaluate algorithm precision and execution time compared to computing polygon similarity through real representations of polygons. Results demonstrated that raster similarity computation is three times faster than exact computation, and raster similarity precision is higher for objects with high similarity and lower for objects that are not very similar. Therefore, we point the use of the proposal when the intention is to process objects that seem to have high similarity. On the other hand, other algorithm must be employed for objects with low similarity, e.g., compute similarity on objects real representation.*

1. Introduction

Quine (1969) and Cakmakov and Celakoska (2004) present the similarity concept as fundamental for learning, knowledge and thought. A similarity metric is a measure that allows comparison of pair of things. Examples of applications where similarity can be used are: medical image databases, human gesture/motion recognition, geologic/geographic information systems, e-commerce, trademark/copyright protection, computer-aided design (Sako and Fujimura, 2000).

Holt (2003) presents spatial similarity as a subset of similarity. It corresponds to a similarity where all the entities being compared to each other have spatial components.

Spatial data consist of points, lines, regions, rectangles, surfaces and volumes (Samet, 1990). Examples of spatial data are: cities, forests, rivers, land use, partition of a country into districts etc. Spatial data is in practice connected to "non-spatial" data (e.g. alphanumeric) (Güting, 1994). Examples of non-spatial data are: names of cities, names of streets, addresses, telephone number etc.

Spatial Database Management System (SDBMS) provides the technology for Geographic Information Systems (GIS) and other applications (Güting, 1994). An important issue in database systems is efficient query processing, and the user receive a query answer in a short time. However, there are many cases where it is not easy to

accomplish this requirement. Besides, there are situations where obtaining fast answers, albeit approximate, is more important to the user than exact ones. This work concerns data compression techniques, i.e., coding mechanisms to generate reduced (or compressed) data over which queries are executed. We are using spatial data signatures to code real data: the Four-Color Raster Signature proposed by Zimbrao and Souza (1998) used to represent polygons.

This work proposes an algorithm to compute similarity of polygons from their 4CRS signatures, named as raster similarity. It employs Jaccard index (Jaccard, 1912) based on the overlapping and common areas of polygons, approximately computed using their raster signatures. Algorithms were implemented in SECONDO, an extensible database that supports non-conventional data types, for example, spatial data (Güting *et al.*, 2005). Experimental tests were conducted on real data corresponding to polygons representing municipalities from north Region of Brazil. The tests were conducted to evaluate the precision of the algorithm and execution time.

This remainder of this work is divided as follows. Section 2 presents the main concepts used in this work. Section 3 presents the proposal, and related algorithms. Section 4 is dedicated to the implementation details and experimental tests, as well as corresponding analysis. Finally, Section 5 presents our conclusions.

2. Theoretical grounding

Approximate Query Processing arises as an alternative to query processing in envirnments for which providing an exact answer results in undesirable response times. The goal is to provide an estimated response in orders of magnitude less time than the time to compute an exact answer by avoiding or minimizing the number of accesses to the base data (Gibbons *et al.*, 1997). Some examples of approximate query processing are: (i) Decision Support Systems, to present aggregate data for decision makers in reasonable time (Hellerstein *et al.*, 1997) (ii) Ad-hoc data mining, during a drill-down query sequence, the earlier queries in the sequence can be used solely to determine what the interesting queries are (Hellerstein *et al.*, 1997). (iii) Spatial OLAP (Online Analytical Processing) (Papadias *et al.*, 2001) to provide fast access to precomputed and summarized data for queries over aggregated data. (iv) Query processing: to provide feedback on how well posed a query is, and even as a tentative answer to a query when the base data is unavailable (Gibbons *et al.*, 1997).

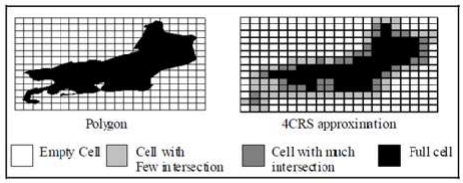
Four-Colour Raster Signature (4CRS) was proposed by Zimbrao and Souza (1998). It is a signature that stores polygon main features in an approximate and compressed representation. The signature can be accessed and processed faster than real data. It corresponds to a grid of cells (Figure 1.b) where each cell store relevant information of object using few bits (Figure 1.a). Grid scale can be adjusted in order to obtain a more compressed representation (lower scale) or a more precise representation (higher scale).

Scale change is used to ensure that signature cells of two 4CRS have same size and that the intersecting cells have the same corner coordinates. One approach to meet this requirement is cells' edge size be a power of two (2^n), and that the beginning of each cell be multiple of the same power of two ($a2^n$), as proposed by Zimbrao and Souza (1998). If this requirement is not accomplished, signatures cells may not overlap as

presented in Figure 2.a, and it is not possible to compare directly polygons signatures. Hence a better approach is perfect overlap of signature cells, as illustrated in Figure 2.b. It is important to emphasize that different signatures can have different cell size. Scale change is accomplished by grouping cells of the signature with smaller cell size, since it is not possible to subdivide a bigger cell to produce smaller ones.

Bit Value	Cell Type	Description
00	Empty	The cell does not intersect the polygon.
01	Weak	The cell contains as intersection of 50% or less with the polygon.
10	Strong	The cell contains an intersection of more than 50% with the polygon.
11	Full	The cell is fully occupied by the polygon.

(a)



(b)

Figure 1. (a) Types of Cell in the 4CRS (Zimbrao and Souza, 1998) and (b) an example of 4CRS (Azevedo *et al*., 2004)

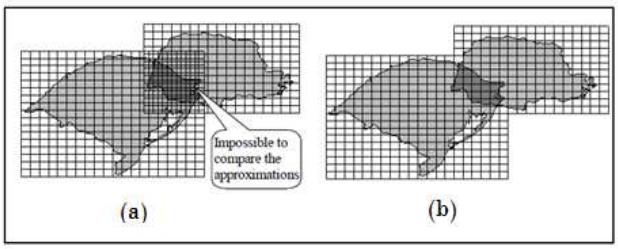


Figure 2. (a) Signatures whose cells do not overlap; (b) Perfect overlap

Zimbrao and Souza (1998) presented good results when 4CRS was used to approximate polygons in exact query processing using the Multi-step Processing of Spatial Joins architecture (Brinkhoff *et al.*, 1994). This motivates the use of 4CRS for approximate query processing, and a set of algorithms was proposed by Azevedo *et al*. (2004, 2005, 2006). These algorithms were evaluated against exact query processing and demonstrated also good results.

Approximate query processing using 4CRS corresponds to, instead of using as input the real object, use object's 4CRS signature, and return an approximate response, along with a confidence interval. As an example, the algorithm that computes the approximate area of a polygon *p* (Azevedo *et al.*, 2004) returns an area value *v*, and an interval *i* with confidence *c*. The response is that real area is between *v-i* and *v+i*, with confidence *c*.

3. Algorithms to Compute Raster Similarity

A similarity function, in an intuitive sense, returns the similarity of objects considering size, shape, and position in space. For instance, for spatial objects that have area, similarity can be computed as the ratio of intersection and union areas, as presented in Equation 1. This equation is an intuitive metric, and it is named as Jaccard index (Jaccard, 1912), as presented by Hemert and Baldock (2007).

$$S(o1,o2) = (A_{\cap}(o1,o2)) / (A_{\cup}(o1,o2)) \quad (1)$$

Where:

- o1 and o2: spatial objects that have area
- $A_{\cap}$: approximate overlapping area of polygons
- $A_{\cup}$: approximate union area of polygons

This work proposes to replace, in Equation 1, *o1* and *o2* by their 4CRS. The algorithm is presented in Figure 3. It has as input the 4CRS of two polygon, and returns a value between the interval [0, 1] that indicates polygons similarity. The algorithm employs other three algorithms: compute approximate area of polygon (Figure 4), compute intersection area of polygons (Figure 5), and compute 4CRS union (Figure 6).

```
REAL similar(signat4CRS1, signat4CRS2)
  intersArea = approxIntersectionArea(signat4CRS1, signat4CRS2);
  IF(intersArea = 0) /* Does not exist intersection area */
    RETURN 0;
  ELSE /* Exists intersection area */
    unionSignat = unionSignat4CRS(signat4CRS1, signat4CRS2);
    unionArea = approximateArea(unionSignat);
  RETURN intersArea / unionArea;
```

Figure 3. Algorithm to compute raster similarity of polygons

The algorithm to compute polygon approximate area (Azevedo *et al.*, 2004) (Figure 4) returns polygon area summing the expected area of the polygon inside each type of signature's cells. The expected areas for *Empty*, *Weak*, *Strong* and *Full* cells are 0%, 25%, 75% and 100%, respectively.

```
REAL approximateArea(signat4CRS)
  nWeakCells = nStrongCells = nFullCells = 0;
  cellArea = signat4CRS.edgeSize * signat4CRS.edgeSize;
  FOR EACH cell IN signat4CRS.cells DO
    IF (cell.type = WEAK)
      nWeakCells= nWeakCells+1;
    ELSE IF (cell.type = STRONG)
      nStrongCells = nStrongCells + 1;
    ELSE IF (cell.type == FULL)
      nFullCells = nFullCells + 1;
  RETURN (nWeakCells * weakWeight + nStrongCells * strongWeight
         + nFullCells * fullWeight) * cellArea;
```

Figure 4. Algorithm to compute approximate area of polygon

The algorithm to compute the approximate overlapping area of two polygons (Azevedo *et al.*, 2005) (Figure 5) sums the expected area of cell types that overlap, and multiplies this value by the cell area. There are four types of cell; hence there are sixteen possibilities of types of cells that overlap, as proposed by Azevedo *et al*. (2005).

```
REAL approxIntersectionArea(signat4CRS1, signat4CRS2)
  interMBR = intersectionMBR(signat4CRS1, signat4CRS2);
  IF (signat4CRS1.edgeSize = signat4CRS2.edgeSize) then
      s4CRS = signat4CRS1;
      b4CRS = signat4CRS2;
  ELSE
      s4CRS = smallerCellSide(signat4CRS1, signat4CRS2);
      b4CRS = biggerCellSide (signat4CRS1, signat4CRS2);
  appArea = 0;
  FOR EACH b4CRS cell b THAT IS inside interMBR DO
      FOR EACH s4CRS cell s THAT IS inside cell b DO
        appArea = appArea + expectedArea[s.type,b.type];
        cellArea = s4CRS.edgeSize * s4CRS.edgeSize;
   RETURN appArea * cellArea;
```

Figure 5. Algorithm to compute overlapping (intersection) area of polygons

The algorithm to compute the signature resulting from the union of two raster signatures is used to compute raster similarity, and it is also a contribution of this work. It computes the signature as follows: if there is intersection MBR (Minimum Bound Rectangle) of the signatures, then a new signature is created and returned. On the other hand, when there is not intersection MBR, then NULL is returned. This simplification was done because if there is no intersection between the signatures, than raster similarity is zero. Some comments help to understand the algorithm (Figure 6). More details about algorithm implementations are presented by Antunes and Azevedo (2011).

```
SIGNAT4CRS unionSignat4CRS(signat4CRS1, signat4CRS2)
  IF existsIntersection(signat4CRS1, signat4CRS2)
    IF (signat4CRS1.edgeSize > signat4CRS2.edgeSize)
      b4CRS = signat4CRS1;
      s4CRS = changeScale(signat4CRS2, signat4CRS1.edgeSize);
    ELSE
      b4CRS = signat4CRS2;
      s4CRS = changeScale(signat4CRS1, signat4CRS2.edgeSize);
      /*unionMBR: MBR that encloses  MBRs of s4CRS and b4CRS */
      unionMBR = computeUnionMBR(s4CRS.MBR, b4CRS.MBR)
      /*Creates 4CRS with Empty Cells */
      n4CRS = createSignature(unionMBR, b4CRS.edgeSize, VAZIO);
      /*Mark each n4crs cell by the union of s4CRS and b4CRS cells*/
      FOR EACH b4CRS cell b that intersects n4CRS cell n DO
        n.type = b.type;
        FOR EACH s4CRS cell s that intersects n4CRS cell n DO
          IF n.type = EMPTY OR s. type = FULL
            n. type = s. type;
          ELSE IF n. type = WEAK AND s. type = STRONG
            n. type = s. type;
      RETURN n4CRS;
  ELSE
    RETURN NULL;
```

Figure 6. Algorithm to compute union of two 4CRS

In approximate query processing, along with the response, it is also important to return a confidence interval. The user can use this interval to decide if the precision of the answer is enough. Equation 2, employed by Azevedo *et al*. (2004, 2005), presents the function to compute the confidence interval for the approximate area and approximate overlapping area algorithms. To execute the calculus, it is required to compute the average and variance of expected area and overlapping expected area.

$$\text{Confidence interval (CI)} = \sum n\ n_c \times [\mu_c \pm p \times \sqrt{(\sigma_c^2/n_c)}] \quad (2)$$

Where:

- c: type of cell or combination of type of cells, according to the algorithm
- μ_c: average (expected area or overlapping expected area)
- σ_c^2: variance
- p: confidence interval, e.g., 1.96 for a confidence interval of 95%
- n_c: number of type of cells.

In this work, we propose a confidence interval for raster similarity, presented in Equation 3, based on the proposals of Azevedo *et al*. (2004, 2005).

$$CI_{\text{Raster similarity}} = [(A_\cap - \triangle CI_\cap)/(A_\cup + \triangle CI_\cup);\ (A_\cap + \triangle CI_\cap)/(A_\cup - \triangle CI_\cup)] \quad (3)$$

Where:

- $A_\cap$: approximate overlapping area of raster signatures
- $A_\cup$: approximate area of union of raster signatures
- $\triangle IC_\cap$: confidence interval variance of approximate overlapping area
- $\triangle CI_\cup$: confidence interval variance of approximate area of union of signatures

An example of confidence interval calculus is presented in Figure 7. Consider that the algorithm execution returned the following data: $A_{\cap}$: 4.95×10^6; $A_{\cup}$: 1.27×10^7; $\Delta CI_{\cap}$: 2.37×10^5; and, $\Delta CI_{\cup}$: 2.45×10^5. Using Equation 3 the confidence interval is:

$$CI_{Raster\ Similarity} = [(4.95 \times 10^6 - 2.37 \times 10^5)/(1.27 \times 10^7 + 2.45 \times 10^5), (4.95 \times 10^6 + 2.37 \times 10^5)/(1.27 \times 10^7 - 2.45 \times 10^5)]$$
$$CI_{Raster\ Similarity} = [(4.72 \times 10^6)/(1.30 \times 10^7), (5.19 \times 10^6)/(1.25 \times 10^7)]$$
$$CI_{Raster\ Similarity} = [0.364, 0.416]$$

Figure 7. Example of confidence interval calculus

4. Experimental Evaluation

4.1. Algorithm implementation

The algorithms were implemented in SECONDO - a generic environment that supports database systems implementation for a large number of data models and query languages (Güting *et al.*, 2005). It is developed as a research prototype at Fernuniversität in Hagen. Implementations in SECONDO are done in algebras. Algebras are based on the concept of second-order signature (Güting, 1993): the first signature describes type constructors and second signature describes operations over these types. As an example, raster similarity operator has the specification presented in Figure 8.

```
Name: rSimilar
Signature: (Raster4CRS, Raster4CRS) -> approxresult
Sintax: _ rSimilar _
Meaning: Returns percent. of similarity between two 4CRS with its confidence interval.
Example: query raster4CRS1 rSimilar raster4CRS2
```

Figure 8. Specification of raster similarity algorithm

All implementations employed in this work are available from the following googlecode project: http://code.google.com/p/raster4crs-project/. The project corresponds to all implementations of Raster Algebra, and the algorithm proposed in this work. In the root directory, there is a readme file that explains how to install this algebra after SECONDO installation. SECONDO is available from http://dna.fernuni-hagen.de/Secondo.html).

4.2. Experimental tests

Experimental tests were performed in order to evaluate the precision and execution time of raster similarity algorithm against polygon similarity computed through real representations of polygons. In the experimental tests, there were used a sample of 382 polygons that represents municipalities from north of Brazil (BRNorth). In order to evaluate raster similarity operator, we generated another data set that overlaps with BRNorth. Original polygons were randomly shifted in the x and y axes, as proposed by Brinkhoff *et al.* (1994), and the data set BRNorthT were generated. Figure 9 presents the data sets. Afterwards, 4CRS signatures were generated for each object of these data sets. All commands used to execute the tests are available in the googlecode project file "Experimental tests of similarity operation".

The next step was to compute the similarity. We collect time of hot execution time. The hot execution time was calculated as the average execution time from a total of 10 executions of the same query, discarding the first, the highest and the slowest times so as to avoid outliers. The time to compute similarity from real polygons was

41.187 seconds, while the time to compute similarity from polygons' 4CRS signatures was 14.406 seconds. Considering this dataset, computing raster similarity is three times faster than computing similarity from real representation of polygons.

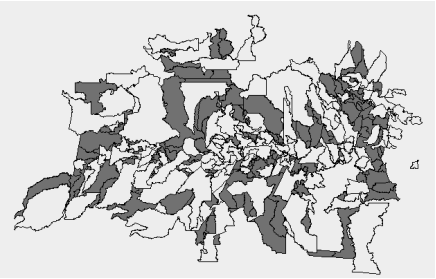


Figure 9. Overlapping of BRNorth and BRNorthT

Afterwards we studied the algorithm precision. We noticed that there were some outliers among the similarity of pair of objects. Objects with very low similarity have big error. Then, to compute the error average the results between percentiles 20 and 80 were considered, excluding extreme values that could bias the average. The error average and error standard deviation of results between percentile 20 and 80 were 13% and 10%. The error median was 10%, and it was used to divide the samples to be studied in two groups: "results above error median (more than 10% of error)" (Table 1) and "results below error median (less than 10% of error)" (Table 2). Due to lack of space, only the most interesting samples for discussion are presented.

The column labels are: (a) ID, IDT: BRNorth and BRNorthT object identifiers; (b) SB, SBT: BRNorth and BRNorthT signatures' length size (size of block); (c) NC: number of cells of signatures that overlap, discarding intersections with *Empty* cells; (d) EIA, AIA: exact intersection area and approximate intersection (overlapping) area; (e) %EAIA: percentage error of approximate intersection area (f) EUA, AUA: exact and approximate union area; (g) %EAUA: percentage error of approximate union area; (h) %RasterS: raster similarity in percentage; (i) %RealS: similarity in percentage computed from real objects; (j) %ES: percentage error of raster similarity (calculated according to Equation 3); (k) %AD: percentage of absolute difference between real similarity and raster similarity (|%RasterS - %RealS|).

$$\%ES = |RasterS - RealS| \,/\, RealS \qquad (3)$$

Table 1. Results above error median

ID	IDT	SB	SBT	NC	EIA	AIA	%EAIA	EUA	AUA	%EAUA	RasterS	RealS	%ES	%AD
234	**397**	**256**	**1024**	**2**	**124**	**312.895**	**252,212.29**	**55,347,900**	**54,263,800**	**1.96%**	**0.58%**	**0.00%**	**257,252.94%**	**0.58%**
10	**25**	**1024**	**256**	**4**	**486**	**1,045,220**	**214,822.12**	**42,117,800**	**41,680,900**	**1.04%**	**2.51%**	**0.00%**	**217,074.82%**	**2.51%**
274	164	512	256	2	1,316	21.286	1,518.03	28,184,300	28,180,500	0.01%	0.08%	0.00%	1,518.24%	0.07%
139	38	256	512	2	9,513	21.286	123.75	18,683,700	17,956,900	3.89%	0.12%	0.05%	132.80%	0.07%
2	**5**	**1024**	**512**	**26**	**5,137,260**	**10,733,300**	**108.93**	**68,657,100**	**68,681,700**	**0.04%**	**15.63%**	**7.48%**	**108.86%**	**8.15%**
140	**146**	**1024**	**1024**	**9**	**613,326**	**1,258,080**	**105.12**	**133,363,000**	**131,596,000**	**1.32%**	**0.96%**	**0.46%**	**107.88%**	**0.50%**

Table 2. Results below error median

ID	IDT	SB	SBT	NC	EIA	AIA	%EAIA	EUA	AUA	%EAUA	RasterS	RealS	%ES	%AD
68	158	512	512	29	2,820,430.00	3,092,300.00	9.64%	34,457,000.00	34,406,400.00	0.15%	8.99%	8.19%	9.80%	0.80%
188	**379**	**1024**	**256**	**22**	**11,475,800.00**	**12,765,200.00**	**11.24%**	**171,566,000.00**	**174,064,000.00**	**1.46%**	**7.33%**	**6.69%**	**9.64%**	**0.64%**
163	38	256	512	12	1,074,720.00	921,305.00	14.27%	17,361,200.00	16,252,900.00	6.38%	5.67%	6.19%	8.43%	0.52%
79	40	128	256	16	389,058.00	424,844.00	9.20%	8,625,600.00	8,716,290.00	1.05%	4.87%	4.51%	8.06%	0.36%
153	**97**	**512**	**512**	**35**	**4,785,020.00**	**4,490,630.00**	**6.15%**	**33,484,000.00**	**34,144,300.00**	**1.97%**	**13.15%**	**14.29%**	**7.97%**	**1.14%**
7	2	512	1024	9	1,770,680.00	1,837,840.00	3.79%	69,856,200.00	67,371,000.00	3.56%	2.73%	2.53%	7.62%	0.19%
79	115	128	256	32	1,036,980.00	1,050,980.00	1.35%	4,741,990.00	4,882,430.00	2.96%	21.53%	21.87%	1.56%	0.34%
280	209	128	256	31	936,733.00	940,258.00	0.38%	7,487,370.00	7,634,940.00	1.97%	12.32%	12.51%	1.56%	0.20%
24	**14**	**1024**	**1024**	**80**	**53,299,100.00**	**52,254,500.00**	**1.96%**	**141,941,000.00**	**141,296,000.00**	**0.45%**	**36.98%**	**37.55%**	**1.51%**	**0.57%**
14	14	1024	1024	61	37,309,900.00	36,984,400.00	0.87%	220,830,000.00	221,512,000.00	0.31%	16.70%	16.90%	1.18%	0.20%
75	32	256	512	13	1,146,480.00	1,181,850.00	3.09%	27,202,300.00	27,721,700.00	1.91%	4.26%	4.21%	1.15%	0.05%
9	9	1024	1024	60	39,526,900.00	39,669,600.00	0.36%	187,091,000.00	188,744,000.00	0.88%	21.02%	21.13%	0.52%	0.11%
434	419	1024	512	28	14,568,800.00	14,089,500.00	3.29%	90,799,300.00	88,080,400.00	2.99%	16.00%	16.05%	0.30%	0.05%
188	**188**	**1024**	**1024**	**163**	**123,756,000.00**	**124,048,000.00**	**0.24%**	**216,988,000.00**	**218,104,000.00**	**0.51%**	**56.88%**	**57.03%**	**0.28%**	**0.16%**
129	228	256	128	52	2,080,950.00	2,136,570.00	2.67%	6,630,980.00	6,799,360.00	2.54%	31.42%	31.38%	0.13%	0.04%
205	146	256	1024	16	5,543,130.00	5,543,720.00	0.01%	74,188,700.00	74,186,800.00	0.00%	7.47%	7.47%	0.01%	0.00%
207	263	128	256	14	306,469.00	307,855.00	0.45%	8,383,900.00	8,421,380.00	0.45%	3.66%	3.66%	0.01%	0.00%

4.2. Result analysis

Raster similarity is computed by the ratio of approximate overlapping area divided by approximate union area. So, it is important to analyze how the overlapping (intersection) area and union area values impact the precision of results. In Figure 10, Y-axis presents percentage of error, while X-axis presents the objects sorted from highest percentage error to lowest percentage error. The percentage error of the area of union of two 4CRS signature is relatively low, while percentage error of raster similarity grows along with percentage error of approximate intersection area. Therefore, if the error of approximate intersection area is small, then the error of raster similarity is small as well.

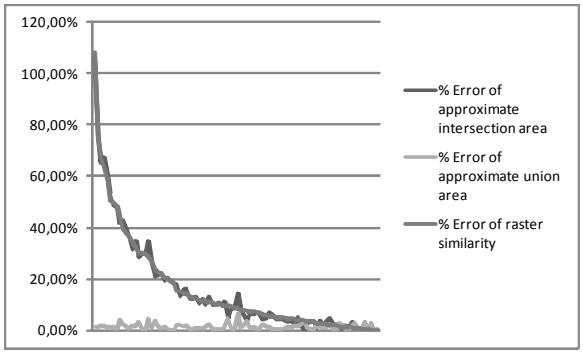


Figure 10. Percentage error of approximate intersection area, approximate union area and raster similarity

The worst error of raster similarity algorithm is presented in the first line of Table 1, corresponding to similarity of objects 234 × 397. The objects are presented in Figure 11.a. In this case, there are only two cells that intersect (column NC). Raster similarity is 0.58% (column RasterS) and real similarity is almost 0% (column RealS). The percentage error is very high (257,252.94%) (column %ES). Similar results occur with objects 10 × 25 (Table 1 and Figure 11.b). It is important to notice that in both examples it is required to execute a scale change of a signature with 256 unities of cell size to a 1024 unities of cell size (columns SB and SBT). The scale change has the goal

to ensure the execution of the algorithm on signatures of same cell size. The scale change is executed grouping a set of cells of the signature with small cell size to represent one cell of the signature with bigger cell size, as presented in Section 2.

Figure 11. Overlapping objects highlighted in Table 1 and Table 2

In the raster similarity calculus of objects 2 × 5 (Table 1 and Figure 11.c), the percentage error of raster similarity is equals to 108.86%. Raster similarity is equals to 15.63% while real similarity is equals to 7.48%. There are more cells that intersect related to the previous example (e.g., NC is equals to 26). However, this number is still small, and it was also required to execute scale change from 512 to 1024. Looking at the type of cells that intersects (*Weak* × *Weak*, *Weak* × *Strong*, *Weak* × *Full*, *Strong* × *Full*, *Strong* × *Strong*, and *Full* × *Full*) (Table 3 - line 1), we can notice that there is no intersection of *Full* × *Full* cells, which is the best case were the precision is 100%.

Table 3. Number of type of cells that overlap

Objects	W × W	W × S	W × F	S × S	S × F	F × F	Total
Obj. 2 × Obj. 5	2	7	6	6	5	0	26
Obj. 188 × Obj. 379	2	1	6	3	6	4	22
Obj. 153 × Obj. 97	3	6	8	2	5	2	26
Obj. 24 × Obj. 14	5	7	18	5	16	29	90
Obj. 188 × Obj. 188	10	9	23	7	33	81	163

On the other hand, in case of raster similarity of objects 140 × 146 (Table 1 and Figure 11.d), there is no scale change, but the number of cells is very small (9 cells). Besides raster similarity and real similarity are very small (0.96% and 0.46%, respectively). Hence the error is 107.88%.

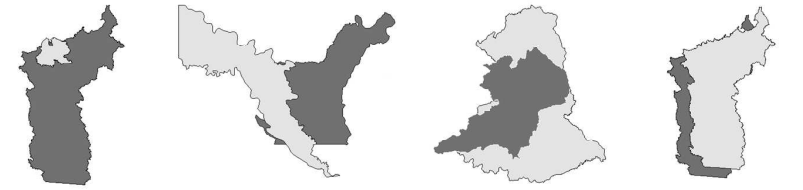
It is important to emphasize that in all cases of Table 1 where the similarity is small, the percentage of absolute difference between real similarity and raster similarity is quite small (column %AD).

We conclude that three main situations contribute to the error: (i) small number of overlapping cells; (ii) majority of overlaps involves cell types whose approximation of overlapping area consider the average (*Weak* × *Weak*, *Weak* × *Strong*, *Weak* × *Full*, *Strong* × *Full*, *Strong* × *Strong*); and, (iii) scale change.

On the other hand, there are other cases where the precision of raster similarity were quite good. For example, in the case of the similarity of objects 188 × 379 (Table 2 and Figure 12.a), the error of raster similarity is 9.64%. Raster similarity is equal to 7.33% and real similarity is 6.69%. The number of cells is small (22 cells), but now there are 4 overlaps of *Full* × *Full* cells (Table 3 - line 2), where the precision is 100%.

In another example, corresponding to similarity of objects 153 × 97 (Table 2 and Figure 12.b), the error of raster similarity is 7.97%. Raster similarity is equals to

13.15%, and real similarity is equal to 14.29%. There are two overlaps of *Full* × *Full* cells (Table 3 – line 3) and the number of cells that intersect is big (35 cells).



(a) Obj. 188 x Obj. 379 (b) Obj. 153 x Obj. 97 (c) Obj. 24 x Obj. 14 (d) Obj. 188 x Obj. 188

Figure 12. Overlapping objects highlighted in Table 3 and Table 4

In the case of similarity of objects 24 × 14 (Table 2 and Figure 12.c), the error of raster similarity is 1.51%. Raster similarity is equals to 36.98% and real similarity is 37.55%. In this case, the number of overlapping cells is 80 and, there are 29 overlaps of *Full* × *Full* cells (Table 3 – line 4).

One of the best results is the similarity of objects 188 × 188 (Table 2 and Figure 12.d). The error of raster similarity is only 0.28%, raster similarity is 56.88% and real similarity is 57.03%. The number of cells that overlap is big (163 overlapping cells) and the majority of overlaps are *Full* × *Full* cells (81 *Full* × *Full* - Table 3 – line 5).

Based on all results presented, we can conclude that the error of the algorithm (highlighted in Figure 10) can be explained because of the approximate intersection area corresponding to the overlap of cell types *Weak* × *Weak*, *Weak* × *Strong*, *Weak* × *Full*, *Strong* × *Full*, *Strong* × *Strong*. When the number of overlaps of these types of cells is small, we cannot assume Normal Distribution, as proposed by Azevedo *et al.* (2004, 2005) to estimate expected area of polygon and expected overlapping area of polygons. Hence two cases can result: (i) When the exact intersection area is close to the average, the approximate result is close to the real result; (ii) When the exact intersection area is not close to the average, the approximate result is also not close to the exact value. This is not the case when Normal Distribution can be applied (Azevedo *et al.*, 2004, 2005). Besides, we observe in our tests that the error above 10% happens when overlapping of objects are on their borders; while in the results with less than 10% of error, objects have more *Full* × *Full* cell overlaps. Therefore, it confirms that the intersection area contributes the most to the error, and it is required to improve the algorithm to compute the approximate intersection area.

Regarding confidence interval, presented in Equation 3 (Section 3), for the results with error above 10%, in 70% of cases, the real similarity was in the interval. In other words, real similarity was between minimum and maximum values computed for the confidence interval presented in Equation 2. On the other hand, for results with error below 10%, in 96% of cases, real similarity was in the confidence interval.

5. Conclusions

The main contribution of this work is a proposal of an algorithm to compute similarity of polygons from their 4CRS signatures. Other contributions were the proposal of algorithm to compute union of two 4CRS signatures and the implementations in SECONDO (Güting *et al.*, 2005) of these two algorithms and algorithms to compute

approximate area of polygons and algorithm to compute approximate overlapping area of polygons (Azevedo *et al.* , 2004, 2005).

Experimental tests were executed over real data corresponding to municipalities from North Region of Brazil. The results demonstrated the proposed raster similarity algorithm is three times faster than the algorithm that computes similarity using real representation of objects. However, raster similarity algorithm's precision varies. Because of some outliers, the percentile 20 and 80 were used to extract a reasonable and interesting sampling for analysis. Among the select objects the median value of error was identified as equals to 10%, and error values below and above 10% was analyzed.

We concluded that the errors above 10% occur when there is small overlapping of objects. The reasons for the error are: (i) small number of overlapping cells of signatures and, consequently, the value of similarity is quite small; (ii) majority of overlapping involves cell types whose approximation consider the average (*Weak* × *Weak*, *Weak* × *Strong*, *Weak* × *Full*, *Strong* × *Full*, *Strong* × *Strong*), which means that the intersection of objects are in their borders; and, (iii) the scale change required to execute the algorithm to compute union of raster signatures. In parallel, the results were quite good for cases where overlaps were bigger.

So we can state that, the bigger is the value of raster similarity, the closer it is to the real similarity. On the other hand, there is a big error in percentage when the value of raster similarity is small. So, if the use of the algorithm intends to discover objects with high similarity, our proposal is a good choice. However, in case of interest is low similarity value, it is better to execute, e.g., the algorithm to compute the real similarity.

We also evaluate our proposal to compute the confidence interval, presented in Equation 3. For the results with error above 10%, in 70% of cases, the real similarity was in the interval. On the other hand, for results with error below 10%, in 96% of cases the real similarity was in the interval. As we employed a 95% of confidence to compute the confidence interval, we can state that our proposal is adequate for high values of raster similarity, but it must be improved for low values of raster similarity.

As future work, we propose: improving the algorithm to compute approximate overlapping area, since the error of raster similarity is highly dependent from overlapping area error, as highlighted in Figure 10; execute performance evaluations considering others datasets; evaluate the use of synthetic data to identify the threshold of similarity for most useful use and recommendation of the algorithm; improve the algorithm to be used in other scenarios, e.g., to compare objects according to their shape, independent from their size and without executing scale change (e.g., compare a model of an object in small size, against a real one); implement a view for Raster objects in SECONDO, which can help to debug the algorithm, and to analyze results.

References

Antunes, L. C. R., Azevedo, L. G, 2011. **"**Polygon Similarity Calculus using Raster Signatures**"**. Technical Report DIA/UNIRIO (RelaTe-DIA), RT-0003/2011, 2011.

Azevedo, L. G., Monteiro, R. S., Zimbrão, G., Souza, J. M. (2004) "Approximate Spatial Query Processing Using Raster Signatures". In: *VI Brazilian Symposium on GeoInformatica* (GeoInfo 2004), Campos do Jordão, Brazil, p. 403-421.

Azevedo, L. G., Zimbrão, G., Souza, J. M., Güting, R. H. (2005). "Estimating the Overlapping Area of Polygon Join". In: International Symposium on Advances in Spatial and Temporal Databases, v. 1, Angra dos Reis, Brazil, p. 91-108.

Azevedo, L. G., Zimbrão, G., Souza, J. M.: (2006). Approximate Query Processing in Spatial Databases Using Raster Signatures. In: Advances in Geoinformatics. 1 ed., v. 1, Springer-Verlag, Berlin Heidelberg New York , p. 69-85.

Brinkhoff, T., Kriegel, H. P., Schneider, R., Seeger, B. (1994). "Multi-step Processing of Spatial Joins". In: ACM SIGMOD Record, v. 23 (2), p. 197-208.

Cakmakov, D., Celakoska, E. (2004). "Estimation of Curve Similarity Using Turning Function". In: Int. Journal of Applied Math, v. 15 (2), p. 403-416.

Gibbons, P. B., Matias, Y., Poosala, V., (1997). "Aqua project white paper". Technical Report, Bell Laboratories, Murray Hill, New Jersey, USA.

Güting, R.H. (1993). "Second-Order Signature: A Tool for Specifying Data Models, Query Processing, and Optimization". SIGMOD Conference, p. 277-286

Güting, R. H. (1994). "An Introduction to Spatial Database Systems". In: The Int. J. on Very Large Data Bases, vol. 3 (4), p. 357-399.

Güting, R. H., Almeida, V., Ansorge, D., Behr T., *et al.* (2005). "SECONDO: An Extensible DBMS Platform for Research Prototyping and Teaching". In: 21st Intl. Conf. on Data Engineering (ICDE), Tokyo, Japan, p. 1115-1116.

Hemert, J. V., Baldock, R. (2007). "Mining Spatial Gene Expression Data for Association Rules". BIRD 2007, LNBI 4414, Springer, p. 66-76

Hellerstein, J. M., Haas, P. J., Wang, H. J. (1997). "Online aggregation". In: Proc. of ACM SIGMOD Intl. Conf. on Manag. of Data, Tucson, Arizona, USA, p. 171-182.

Holt, A. (2003). "Spatial similarity". In: 15th Annual Colloquium of the Spatial Information Research Centre, Dunedin, New Zealand, p. 77-80.

Jaccard, P. (1912). "The distribution of flora in the alpine zone". In: The New Phytologist, vol. 11(2), p. 37–50.

Papadias, D., Kalnis, P., Zhang, J. *et al.* (2001). "Efficient OLAP Operations in Spatial Data Warehouses". In: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, Redondo Beach, CA, USA, p. 443-459.

Quine, W. V. (1969). "Ontological Relativity and Other Essays". In: Columbia University Press, New York.

Sako, Y., Fujimura, K. (2000). "Shape Similarity by Homotopic Deformation". In: The Visual Computer, vol. 16(1), p. 47-61.

Samet, H. (1990). "The Design and Analysis of Spatial Data Structure". Addison-Wesley Publishing Company, 1st edition.

Zimbrao, G., Souza, J. M. (1998). "A Raster Approximation for Processing of Spatial Joins". In.: Proceedings of the 24rd International Conference on Very Large Data Bases, New York City, New York, USA, p. 558-569.