Proc. of the Intl. Conf. on Advances In Information Processing And Communication Technology - IPCT 2014.

ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-83

Proposal for computing areas of polygons in Geographic Information System. Computational Analysis

Romanuel Ramón Antunez, Lidisy Hernández Montero

***Abstract*— Most of spatial analyzes in geographic information systems (GIS) use simple geometric calculations, from which more complex algorithms are built. Traditionally these calculations have been conducted in Euclidean metric, but the geographical information has become more available and planimetric accuracy what has caused these calculations in GIS are inaccurate. These errors could become significantly important in some individuals cases, such as large geographical areas or equidistant projections. Therefore in this paper we proposed an algorithm for calculating the area of polygons supported in a generic method that takes into account the coordinate system and / or projection with which it works, allowing a correct and efficient functioning in GIS applications.**

***Keywords*—algorithm, computational geometry, geometric analysis, geographic information systems.**

I. Introduction

Most of the information handled in business and / or government has a close relationship with spatial data, for that reason decision making and the precision thereof, are conditioned largely by the quality, accuracy and currency of this spatial information. Then turning the Geographic Information System (GIS) in vital tools for support in those decisions.

One of the computing disciplines that presenting their contributions in a fairly immediate way in these systems is the Computational Geometry.

This discipline provides criteria for detecting and arranged geometric structures as well as its digital representation. Develop computational tools for the analysis of geometric problems and proposes strategies for implementing efficient algorithms that facilitate the effective resolution of these computational problems (Coelho, 2004).

Usually in this discipline the treated and applied in GIS problems have been worked as Euclidean metric. Moreover geographic information has gained in availability and planimetric accuracy what has caused these calculations in GIS are inaccurate (Pesquer, et al., 2003).

Romanuel Ramón Antunez
Instituto Superior para as Tecnologías da Informação e Comunicações
Angola/Cuba
romanuelr@gmail.com

Lidisy Hernández Montero
Universidad de las Ciencias Informáticas
Cuba
lhernandez@uci.cu

For proper performance of these functions is essential to analyze the quality of the map databases involved, the characteristics of the reference system, basically projection and ellipsoid or spherical model, the dimensions of the area and objectives, resolution of the resulting bases, planimetric accuracy , among others. Based on these parameters, the GIS analysis tools should facilitate to the non-expert user in these issues a valid solution automatically, and the advanced user adequate control to enforce their decision to use an Euclidean's methods , or other. It would be absurd to force the user to use always and in any condition, the more accurate but much slower methods of geodesy (Pesquer, et al., 2003).

In the context of this paper, is proposed a general method for geometric calculations, taking into account the coordinate system and / or projection with which it works; and specifically the development of a new algorithm based on spherical geometry to determine the area of spherical polygons without using geodetic models for re-projection and / or eliminating restrictions on the digitization of these polygons.

II. Solution Overview

A. *A Generic Method*

A map projection can be defined as a bijection; then by theorem is that if f is a bijective function, then f¯1 exist and there is also bijective.

Based on this theorem it is clear that when working with projected coordinate is possible to obtain the corresponding ellipsoidal coordinates on the reference ellipsoid which it departed.

The proposed generic method is based on this property and taking into account that in a geographic information system is always known projection and / or coordinate system in which you are working, the idea presented is as follows:

- If is working in a system projected then it can work with algorithms for planar coordinates or algorithms for ellipsoidal coordinates with prior flat rectangular transformation of ellipsoidal coordinates.
    - This decision can be taken by the user in case of be expert; default work converting the ellipsoidal coordinates.
- In case not be working in a projected system then work with algorithms for the ellipsoidal coordinates.

Generic method in pseudocode would be as follows:

Proc. of the Intl. Conf. on Advances In Information Processing And Communication Technology - IPCT 2014.

ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-83

```
function X (x: type)
    if projection do
        if cartesian do
            return CartesianX(x: type)
        endif
        return SphericalX(Transform(x: type))
    endif
    return SphericalX(x: type)
endfunction
```

The projection variable is a boolean would be true if being worked with a map projected and false in other case, while the cartesian be another boolean variable would be initialized to false by default to work with the algorithm for the ellipsoidal coordinates, makes the proposed to work always in ellipsoidal coordinates because the fact of working with a map projection does not guarantee that the results obtained by plane geometric functions are correct, it must be taken into account for what purpose this projection was built and which therefore properties are maintained, however the value of this variable can be changed at the user depending on the knowledge of the projection with which you are working and depending on interests.

B. Algorithm for calculating areas

1) Algorithm for the area of simple polygons in the plane

One of the most widely used algorithms to find the area of a polygon in the plane is based on the incremental technique. Consists from a vertex initially , trace diagonal to the remaining vertices, obtaining a triangulation of the polygon, then the area of the polygon in question would be the sum of the areas of the resulting triangles.

Find the area of a triangle in the plane, is a trivial problem resolution, as would be given by the cross product divided between two of the segments p1p2 and p2p3. Where p1,p2 and p3 are the vertex of triangle (Chen, 1996).

This would give the resulting area with a positive sign if the vertices are taken anticlockwise and expressed in correspondence with the unit of measure in which they are expressed p1, p2 and p3

This algorithm has a complexity linear O(n) as shown in the pseudocode:

```
function LeftTurn( pto1, pto2, pto3: Point)
    return (pto1.x * pto2.y) – (pto1.y * pto2.x) + (pto2.x *
        pto3.y) – (pto2.y * pto3.x) + (pto1.x * pto3.y) –
        (pto1.y  * pto3.x)
endfunction
```

```
function SurfaceCartesianPolygon (p: Polygon)
  area ← 0
  for i ← 0 to p.vertex – 2 step 1
     area←area+LeftTurn(p.vertex[0],p.vertex[i],
                        p.vertex[i+1])
  endfor
  if area < 0 do
    return (area * -1) / 2
  endif
  return area / 2
endfunction
```

In the algorithm presented all areas of the triangles are calculated double and divided only at the end, to thereby reduce truncation errors that may exist to perform calculations on a computer due to rounding.

2) Algorithm for the area of simple polygons spherical

In the case of calculating the area of simple polygons on the sphere, in [Bevis, et al, 1987] is proposed an algorithm based on the formulation of the area of spherical polygons.

That formula only depends on the amplitude of the interior angles of the polygon, and it is this breadth which attempts to calculate the algorithm presented by (Bevis et al., 1987). However to ensure that they accurately measure the interior angles of a restriction imposed in this algorithm, and the need for the polygon is scanned clockwise, otherwise should be sorted polygon vertices obtaining an upper bound of complexity for the algortim igual O(nlogn).

For this reason another variant is designed to determine the area of these polygons by removing this restriction on the digitization and computationally less expensive. The new design philosophy remains the same incremental case for polygons in the plane with changes in how to calculate the area of triangles.

The algorithm takes as input only the polygon to which you want to determine the area. First, a vertex is selected as initial, and from this the remaining vertices geodesic lines are drawn. In this way a set of spherical triangles are obtained. The next step would be to add or subtract the area of each of these triangles depending on the rotation performed to explore its vertices.

In order to calculate the area of these triangles, the lengths of its sides are calculated by Haversine. Then the cosine formula for spherical geometries are each of the angles and finally with these values can be get the area of these triangles.

```
function Fcoseno (l1, l2, l3: double)
    cose ← (cos(l1)  –  cos(l2) * cos(l3)) / (sin(l2) * sin(l3))
    return arccos(cose)
endfunction
```

Proc. of the Intl. Conf. on Advances In Information Processing And Communication Technology - IPCT 2014.

ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-83

```
function SphericalDistance(_pto1, _pto2: Point)
  _radio ← 6 378 400
  _dlon ← _pto2.x - _pto1.x
  _dlat ← _pto2.y - _pto1.y
  _intcal ← pow(sin(_dlat/2),2) + cos(_pto1.y) * cos(_pto2.y)
            * pow(sin(_dlon/2),2)
  _intd ← 2 * arcsin(min(1.0,sqrt(z)))
  return _intd
endfunction

function Fcoseno (l1, l2, l3: double)
   cose ← (cos(l1) – cos(l2) * cos(l3)) / (sin(l2) * sin(l3))
   return arccos(cose)
endfunction

function SurfaceSphericalTriangle (pto1, pto2, pto3: Point)
  radio ← 6 378 100
  l1 ← SphericalDistance (pto1,pto2)
  l2 ← SphericalDistance (pto2,pto3)
  l3 ← SphericalDistance (pto1,pto3)
  a1 ← Fcoseno(l1,l2,l3)
  a2 ← Fcoseno(l2,l3,l1)
  a3 ← Fcoseno(l3,l1,l2)
  epsilon ← (a1 + a2 + a3) – Pi
   area ← epsilon * pow(radio, 2)
  if LeftTurn(pto1, pto2, pto3) < 0 do
     area ← area * -1
  endif
  return area
endfunction

function SurfaceSphericalPolygon (p: Polygon)
  area ← 0
  for i ← 0 to p.vertex – 2 step 1
area←area+SurfaceSphericalTriangle(p.vertex[0],p.vertex[i],
                                   p.vertex[i+1])
endfor
if area < 0 do
    return area * -1
endif
  return area
endfunction
```

Of this form is maintained the same order of complexity O(n) as the case of planar polygons.

3) Pseudocode of the Generic Method for the Area

Now once it have the algorithms to the area of polygons both plane and spherical, can be passed to generic design linking the two algorithms.

```
function CalculateSurface(p: Polygon)
  if projection do
    if cartesian do
       return SurfaceCartesianPolygon(p)
    endif
    return SurfaceSphericalPolygon(Transform(p))
  endif
  return SurfaceSphericalPolygon(p)
endfunction
```

III. Analysis of Results

A. Analysis of the complexity of the algorithm for the area

The proposed generic method for the area calculation are carried out two comparisons of complexity O (1) depending on the outcome are execute auxiliary functions which would dictate the overall complexity of the algorithm running. So it then proceeds to analyze the complexity of each of these functions.

1) Analysis of the complexity of the function SurfaceCartesianPolygon

The SurfaceCartesianPolygon function implements a loop that repeats n-2 times the LeftTurn function, and this function at the same time are executed only elementary operations sum, subtraction and multiplication, so it has a complexity O (1). then obtaining the following theorem:

Theorem 1: The area of a polygon in the plane can be computed with a complexity O (n). Where n is the number of polygon vertices.

2) Analysis of the complexity of the function SphericalDistance

As shown SphericalDistance function does not implement its own cycle and a set of assignments and mathematical calculations, so its complexity can say a priori that it would be O (1). However, one should analyze the case of square root and trigonometric functions used in that role.

In (Brent, 1976), it posed that all these functions can be computed in constant time depending on the level of precision to be obtained. And there came the lemma:

Proc. of the Intl. Conf. on Advances In Information Processing And Communication Technology - IPCT 2014.

ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-83

Lemma 1: The distance between two points on the sphere can be computed with a complexity O(h). Where h is a constant that varies depending on the precision to be obtained.

3) Analysis of the complexity of the function SurfaceSphericalPolygon

In the function SurfaceSphericalPolygon follows the same design that in the function SurfaceCartesianPolygon raised , for this reason a priori one could say that this function has a complexity O (n) by Theorem 1.

However, this function using trigonometric functions become so this dimension is unchanged by Lemma 1 being then:

Theorem 2: The area of a spherical polygon can be computed with a complexity O(nh).

4) Analysis of the complexity of the generic algorithm for the area

Now after analyzing the complexity of the auxiliary functions used in the preparation of generic method for calculating the area can proceed to analyze the complexity.

Would have a complexity O (n) for the case you are working with projected coordinates and want to use a method for the plane, by Theorem 1 and if you want to use a method for ellipsoidal coordinates would . complexity O (nh) by theorem 2. Finally we have:

Theorem 3: The area of a polygon may be computed, taking into account the coordinate system in which it is, with a linear complexity.

This shows that the proposed algorithm for calculating the area of spherical polygons can compute linearly without imposing the sorting restriction posed by (Bevis et al., 1987).

B. Practical Tests

To verify that the results in a real practice correspond to the theoretically obtained results, a set of tests were made to compare what difference there was among the results of the proposed algorithm for the calculation of spherical polygons area, and the results shown by the algorithms currently implemented in geographic information systems with open and close source code respectively, Fig. 1.

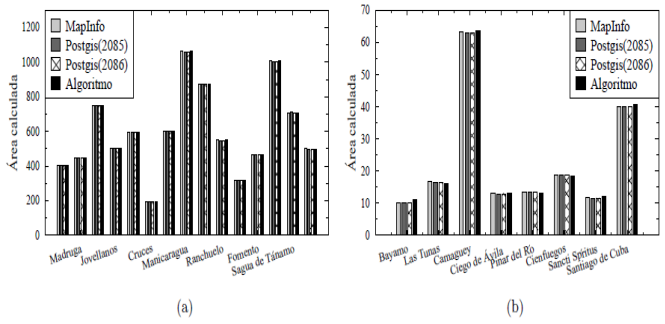

Figure 1. Obtained results over scale mapping based on 1:500 000 (a) and 1:100 000 (b).

These tests were performed with geometries at different scales, as geographic information can be analyzed at different levels and depending on the level used, the results may be of different nature (Ramón, 2011) (Mandelbrot, 1983). It is verified that the results shown by the proposed algorithm are comparable to the classic variants, while reducing the response time to requests since the calculations are performed in O(n) in any case.

IV. Conclusions

- The geometric analysis in GIS is not a simple problem if that want to get valid results. The user must be aware of the nature of the data with that are working , what kind of projection and algorithms used depending on the results that want to obtain.
- The approximation of the spheroid / ellipsoid to a sphere allows obtain simple and valid algorithms, with minimal sacrifice in the accuracy of the results, for scales larger than 1:100,000 for geometric calculations on their surface.
- Using geodetic algorithms is eliminated when working in geographic coordinates, yielding linear complexity algorithms, without the need for restrictions on the digitization of geometries.
- The combination of these algorithms with the Euclidean classical, taking into account the characteristics of the data, lets give advanced users the option to choose what type of method he want to work. And inexperienced users in cartography and geodesy issues guarantees to obtain valid results independent of the coordinate system he are working with.

V. References

[1] Aguirre G. N., Ortíz M. A., Nernand. O. *Principios Básicos de la Cartografía Temática*. Bogotá Colombia, IGAC. ISBN: 958-9067-32-8, 1998.

[2] Bevis M., Cambareri G. *Computing the Area of a Spherical Polygon of Arbitrary Shape*. Mathematical Geology. 1987, Vol. 19, 4. 1987.

[3] Chen, J. *Computational Geometry - Methods and Applications*. Texas: A&M University, Computer Science Departament, 1996

[4] COELHO DE PINA, JOSÉ. Instituto de Matemática e Estadística. 2004. [en línea] http://www.ime.usp.br/~cris/mac331/intro.pdf

[5] CONTRERAS-ALCALÁ, FELIPE. Cutting Polygons and a Problems on Illumination stages. Dept. Compt. Ottawa Ont, Canada: Sci. University of Ottawa, 1998. Master's Thesis.

[6] GALO, MAURICIO, GALERA MONICO, JOÃO F. y CASTRO DE OLIVEIRA, LEONARDO. Cálculo de Áreas De Polígonos Sobre o Elipsóide Usando Projeções Equivalentes. s.l.: Anais do III Colóquio Brasileiro de Ciências Geodésicas.

[7] GOIZUETA, JAVIER. Sistemas de Referencia Geodésicos. Proyecciones Cartográficas. s.l.: ECAS Técnicos Asociados S.A., 2006.

[8] Haining . R. *Spatial Data Analysis: theory and practice*. Cambridge University Press, ISBN: 0-521-7743-73, 2003.

[9] Mandelbrot .B.B. *The Fractal Geometry of Nature*. Henry Holt and Company, ISBN: 0716711869, 1983.

ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-83

[10] Mitchell, Tyler. *Web Mapping Illustrated: Using Open Source GIS Toolkits.* s.l. : O´Reilly, 2005. ISBN: 9780596008659.

[11] Olaya, V. *Sistemas de Información Geográfica.* s.l. : OSGEO:Your Open Source Compass, 2011. Pág 911.

[12] PESQUER MAYOS, LLUÍS, PONS FERNÁNDEZ, XAVIER y MASÓ PAU, JOAN. International Society for Photogrametry and Remote Sensing. [en línea] 2003. Disponible en: [http://www.isprs.org/publications/related/semana_geomatica05/front/abstracts/Dimarts8/G08_abs.pdf].

[13] Ramón R.: *PROPUESTA DE ALGORITMOS PARA ANÁLISIS GEOMÉTRICOS EN SIG*. Master's thesis, Departamento de Geoinformática, Facultad 6, Universidad de las Ciencias Informáticas, diciembre 2011.

[14] RICHARDUS, P. y ADLER, R. K. Map Projection from Geodist, Cartogrphers and Geographers. Amsterdam: North-Holland, 1972.

[15] ROBINSON, A. H., et al. Elements of Cartography. New York: John Wiley and Sons, 1984.

[16] RODRIGUEZ ROCHE, ERNESTO. El problema de la transformación de coordenadas determinadas con GPS. La Habana, Cuba: Departamento de Geodesia y Cartografía. GEOCUBA Investigación y Consultoría, 2009.

[17] S. CHINEA, CARLOS. Las fórmulas de la trigonometría esférica. 2002.

[18] SEDGEWICK, ROBERT. Geometric Algorithms. Algorithms. Menlo Park, California: Addison-Wesley Publishing Company, Inc., 1983.

[19] SNYDER, J. P. Map Projection - A Working Manual. Washington: United State Government Printing Office, 1987.

[20] Tomlin, C.D. Geographic information systems and cartographic modelling. Prentice Hall., ISBN: 0133509273, 1990.

[21] Wood. J. The Geomorphological Characterization of Digital Elevation Models. Ph.D. thesis, University of Leicester, 1996.