Metamorphic Testing of Spatial Distance Measuring Function of GIS

Song Huang, Yiting Duanmu, Zhanwei Hui, Mengyu Ji
PLA Software Test and Evaluation Centre for Military Training, Nanjing, China
PLA University of Science and Technology, Nanjing, China
E-mail: hs0317@sohu.com, duanmuyiting@yahoo.cn, hzw_1983821@163.com, txmxr@163.com

***Abstract*—Spatial distance calculation is a basic and primary function of GIS, however due to the complication of the algorithm and the existence of rounding and truncation errors, it is very difficult to obtain the oracles of the implementation, so traditional testing methods face great challenges. Metamorphic testing can help solve the oracle problem by comparing the relations among multiple inputs and outputs. This paper tries to find out the error in spatial distance measuring of GIS using metamorphic testing. We choose a universal algorithm called the Great Circle Calculation, produce 5 metamorphic relations for it and then carry out the metamorphic testing. The experimental results show that metamorphic testing is effective in finding errors of spatial distance measuring.*)**

Keywords-Metamorphic testing, GIS, Spatial distance calculation

I. INTRODUCTION

Nowadays, Geographic Information Systems (GIS) are applied widely in military training software systems. The spatial measuring function is a main part of GIS, it consists of spatial space calculation, spatial orientation calculation, area calculation, volume calculation, gradient calculation and so on [1]. In theses GIS applying military software, spatial measuring function plays an important role in analyzing war field terrain information quantitatively, which provides necessary information for making campaign plans, forecasting trend of the battle and armament production, all of these provide theoretical basis for making military commands and decisions [2]. So quality of spatial measuring results needs to be assured for military training software.

Spatial distance calculation refers to calculating the distance between two spots on earth and it is one of the most basic parts of spatial measuring of GIS and its correctness needs assurance. Formal proofs of the algorithm's correctness do not guarantee that an application implements or uses the algorithm correctly, so software testing is needed [3]. Spatial distance calculation is a kind of numerical program in its essence and the major feature of scientific computing is "approximation" [4], oracle problems exist in its testing.

This paper studies metamorphic testing in testing spatial distance calculation. We study the Great Circle Calculation algorithm, analyze its testing challenges, and conduct experiments to examine metamorphic testing's effectiveness in detecting faults. Section 2 introduces the metamorphic testing technique and the oracle problem of testing spatial distance calculation. Section 3 explains the algorithm of spatial distance calculation and its implementation -DOP, and import a mutation to obtain a fault version called EDOP. Section 4 depicts the whole testing procedure and section 5 gives the conclusion.

II. BACKGROUND

A. Metamorphic Testing.

Metamorphic testing technique [5] is designed as a general technique for creating follow-up test cases based on existing ones, particularly those that have not revealed any failure, in order to try to find uncovered flaws. The follow-up test cases are created based on "metamorphic relations", and the outputs from these test cases can then be predicted, if the real outputs deviate from the predicted ones, then faults exist in the program tested.

Metamorphic relation refers to the relations among several inputs and their corresponding outputs which must be satisfied if the program is correct. The relations can be any kind including equality, convergence and so on. Suppose program P is the implementation of function f, x1, x2,…xn （n>1）are inputs which output f(x1)，f(x2)，…，f(xn). Metamorphic relation of f can be described as follows:

$$r(x_1, x_2, \ldots, x_n) \Rightarrow r_f(\mathrm{f}(\mathrm{x}_1), \mathrm{f}(\mathrm{x}_2), \ldots, \mathrm{f}(\mathrm{x}_n)) \quad (1)$$

The key part of metamorphic testing is to check the satisfaction of metamorphic relations.

Here is an example of metamorphic testing. Suppose program P(a, b) is the implementation of integral function $f(a,b) = \int_a^b f(x)dx$, the metamorphic relation we construct is $\int_a^c f(x)dx + \int_c^b f(x)dx = \int_a^b f(x)dx$, then we obtain three test cases $P(a,c)$, $P(c,b)$ and $P(a,b)$. If the program is correct, then the outputs must satisfy the equation that $P(a,c) + P(b,c) = P(a,b)$, if the equation is not satisfied, then there must be faults in the program.

B. Spatial Distance Calculation and the Oracle Problem for Its Testing.

Spatial distance calculation uses the coordinates of two points and outputs the distance between them according to certain algorithm. There are three ways of calculation including plane calculation, spatial calculation and sphere calculation which are used in different situations. As for the fact that GIS handles real geographical data, the last one is used most widely.

Distance calculation involves scientific computing, and its results can be affected by many factors such as the precision of the empirical data collected in laboratories, rounding errors, truncation errors and the quality of the mathematical model, all of these make the computing results

Published by Atlantis Press, Paris, France.

an approximation. Generally speaking, the oracle of a numerical program is obtained from another version of program that implements the same function. However, such program is particularly hard to acquire and may also suffer from the tendency to contain similar faults in program under test. Another way to get oracle is tabular data accumulated in the literature, but the data is extremely limited and is not sufficient enough for testing. All the above factors result in the oracle problem for the testing of distance calculation.

III. SPATIAL DISTANCE CALCULATION ALGORIGHTM AND ITS IMPLEMENTATION

A. *The Algorithm of Great Circle Calculation.*

The earth is an ellipse sphere and distance between two points can't be calculated by Euclidean distance, instead the great circle arc length is used to denote the spherical distance between two points [1]. To obtain the great circle, we intersect a plane crossing the centre of sphere with the surface of the earth, the arc length refers to the shorter arc on the great circle with two points on it. The sphere calculation algorithm calculates the length of the great circle arc and is one of the most popular distance calculation algorithms. Famous GISs such as MapInfo and ArcGis realize the distance calculation on the basis of this algorithm.

Here is how the sphere calculation algorithm works:

Suppose there are two geographical points on the planet, they are P(w1,j1) and P(w2,j2) respectively, where w refers to the latitude and j refers to the longitude. According to the triangle cosine theorem, the cosine of arc P1P2 can be calculated with formula (2):

$$\cos\partial = \cos w_1 \bullet \cos w_2 \bullet \cos(j_1 - j_2) + \sin w_1 \bullet \sin w_2 \quad (2)$$

The distance between P1 and P2 can be calculated with formula (3), where R is the radius of the earth:

$$P_1P_2 = R \bullet \arccos(\cos w_1 \bullet \cos w_2 \bullet \cos(j_1 - j_2) + \sin w_1 \bullet \sin w_2) \quad (4)$$

B. *Implementation of Sphere Distance Calculation Algorithm.*

We present the implementation of sphere distance calculation of ArcGIS in Fig.1.

There are two functions, Distance Of Two Points is the main function and Rad is called to transform degree to radian. The inputs of the main function include the coordinates of two points and projected coordinate system and the output is the distance. The parameter of projected coordinate system is used to determine the radius of the sphere, in this paper we choose a fixed one WGS84, which means the radius is 6378137.0, and we call this implementation DOP. The inputs of the implementation are (w1, j1) and (w2, j2), where w is the latitude and j is the longitude. The output of DOP represents as P((w1,j1), (w2,j2)).

$$P((w_1, j_1), (w_2, j_2)) = 0, (w_1 = w_2 \ \&\ y_1 = y_2) \quad (4)$$

```
1) public static double DistanceOfTwoPoints(double
lng1, double lat1, double lng2, double lat2,
GaussSphere gs)
3){  double radLat1 = Rad(lat1);
4)   double radLat2 = Rad(lat2);
5)   double a = radLat1 - radLat2;
6)   double b = Rad(lng1) - Rad(lng2);
7)   double s = 2 *
Math.Asin(Math.Sqrt(Math.Pow(Math.Sin(a / 2), 2)
+ Math.Cos(radLat1) * Math.Cos(radLat2) *
Math.Pow(Math.Sin(b / 2), 2)));
8)   s = s * (gs == GaussSphere.WGS84 ?
6378137.0 : (gs == GaussSphere.Xian80 ?
6378140.0 : 6378245.0));
```

Figure 1. Distance calculating program in ArcGIS-DOP

IV. METAMORPHIC TESTING OF SPATIAL DISTANCE CALCULATION

A. *Oracle Problem of Spatial Distance Calculation Testing.*

First we create a special test case for DOP. Considering the algorithm's feature to calculate distance between two points, we can easily obtain test cases of (4).

It shows that when the coordinates of two input points are totally same, then the output shoud be zero, which means that the distance between the same points is zero. When it comes to test cases generation, we choose a random way. First determine the interval of latitude and longitude which are (-90，90) and (0，360) respectively, then generate the coordinates randomly. Combining testers experience with testing, we end up with the following five test cases: C1((0,0), (0,0)), C2((35,160), (35,160)), C3((-26,78), (-26,78)), C4((-56,230), (-56,230)),C5((89, 359), (89,359)). Then we execute them with EDOP, all of the outputs are 0, no errors are found.

Then we try to use pseudo-oracles to test DOP. We create a test case C((0,0), (0,180)), then use distance calculation Distance()[6] of MapX as a reference program to execute the test case, the output is 20015077.371417216 meters, while the outputs of EDOP under three projected coordinate systems are 20037847.6348 meters, 20037517.7676 meters and 20037508.3428 meters. Analyzing all the results we find that all of the three outputs of EDOP are different from that of Distance, however the biggest difference is no more than 0.11%, which is quite subtle, considering the inherent approximation feature of scientific computing, the differences can't guarantee the existence of errors.

In this section we find that testing of spatial distance calculation is lack of oracles, thus traditional testing methods face great challenges.

B. *Construction of Metamorphic Relations.*

Though special case testing is not efficient enough, we can still use it as a supplement to metamorphic testing. In the following part we will discuss how to apply metamorphic testing to EDOP.

A challenge of metamorphic testing is to determine proper metamorphic relations. We need to analyze the necessary properties of the program and construct metamorphic relations. There have been some researches on how to devise metamorphic relations [7]. Enlightened by these researches, we study the features of the algorithm. First some properties of trigonometric function can be used and some geometrical properties about distance between two points can also be considered, then finally we choose the following 5 relations:

$$R_1 : P((w_1, j_1),(w_2, j_2)) = P((w_1 + \pi, j_1),(w_2 + \pi, j_2))$$
$$R_2 : P((w_1, j_1),(w_2, j_2)) = P((w_1, j_1 + \pi),(w_2, j_2 + \pi))$$
$$R_3 : P((w_1, j_1),(w_2, j_2)) = P((-w_1, j_1),(-w_2, j_2))$$
$$R_4 : P((w_1, j_1),(w_2, j_2)) = P((w_2, j_2),(w_1, j_1))$$
$$R_5 : P((w_1, j_1),(w_2, j_2)) = P((-w_1, j_1 + \pi),(-w_2, j_2 + \pi))$$

In the formulae above, value of π is 180 when it comes to calculation.

V. EXPERIMENT

A. Mutation Importation.

By analyzing DOP we can find that code in line 7 is core of the program, errors in this part would severely affect the functionality of the program, and should be tested thoroughly. To examine the effectiveness of metamorphic for subtle errors, we insert a mutation in line 7:

```
7) double s = 2 * Math.Asin(Math.Sqrt(Math.Pow(Math.Sin(a / 2), 3) + Math.Cos(radLat1) * Math.Cos(radLat2) * Math.Pow(Math.Sin(b / 2), 2)));
```

The italic "3" is where the mutation seeded and it was 2 originally. The mutated version of DOP is called EDOP. The error we seed is very subtle and can be neglected easily, because it would not result in evidently wrong output. Metamorphic testing is good at detecting such errors. In the following part we are going to test EDOP with metamorphic testing technique.

B. Generation of Original and Follow-up Test Cases.

The interval of latitude is (-90, 90) and of longitude is (0, 360). We generate the coordinates of points randomly based on the intervals. We generate original test cases randomly and they are represented with Co. Then follow-up test cases are generated based on Co for every metamorphic relation, they are represented with Cf. For example one original test cases is Co(30, 60), the metamorphic relation to be tested is R1, then we would obtain a follow-up test case Cf(210, 240).

8 original test cases we generate are shown in Table1.

TABLE I. ORIGINAL TEST CASES GENERATED

Original Test Cases	(w1, j1)	(w2, j2)
Co1	(-51.8, 76.3)	(-50.0, 79.9)
Co2	(-14.6, 150.6)	(44.3, 268.7)
Co3	(33.8, 247.7)	(-52.8, 74.4)
Co4	(22.9, 225.9)	(0.5, 181.1)
Co5	(-16.9, 146.3)	(6.6, 193.2)
Co6	(-74.1, 14.9)	(-52.5, 102.6)
Co7	(67.5, 7.0)	(12.6, 256.3)
Co8	(10.5, 200.9)	(-13.6, 152)

Test Cases Execution and Results Comparison. There are 8 original test cases and 5 metamorphic relations, so 40 pairs of test cases need to be executed and their results to be compared. After executing these test cases with EDOP, we then examine whether the results satisfy the corresponding metamorphic relation or violate it, results are shown in Table2, where P means Pass and F means Fail.

TABLE II. RESULTS OF OUTPUTS COMPARISON

No.	R1	R2	R3	R4	R5
Co1	P	P	F	F	F
Co2	P	P	F	F	F
Co3	P	P	F	F	F
Co4	P	P	F	F	F
Co5	P	P	F	F	F
Co6	P	P	F	F	F
Co7	P	P	F	F	F
Co8	P	P	F	F	F

By analyzing the results, we find that metamorphic testing detects errors in EDOP effectively. 3/5 of the test cases can cause failure, which reports the existence of faults. Besides, different relations show different abilities to detect the mutation we seed. Among the 5 relations, R1 and R2 are obtained merely from formula (3) based on cosine law, while R3, R4 and R5 are devised based on the geometrical properties of the algorithm, such as the distance between the symmetrical ones about the equator of the original points should be the same. Experimental results show that the latter metamorphic relations do a better job in finding errors possibly due to its closer relationship with the algorithm.

VI. SUMMARIES

We study how to apply metamorphic testing in testing spatial distance calculation function of GIS with the existence of oracle problems. Experimental results show that metamorphic testing is an effective in detecting subtle errors while there are no effective oracles, this provide fundamental contributions to testing other spatial calculation function of GIS. However the testing method proposed needs to be improved. Firstly, the construction and selection of metamorphic relations need further researches. The metamorphic relations in this paper are devised with great subjectivity and will be affected by experiences of testers. Secondly, automation of such metamorphic testing procedure is strongly needed for test case generation, execution and results comparison, testing wil be low efficient when the amount of data is large which is quite often in GIS testing. At last, program information should be considered to generate original test cases to improve the pertinence, so we would try to find new test case generation methods.

ACKNOWLEDGMENT

This work was supported by National High Technology Research and Development Program of China Project (No: 2009AA01Z402).

REFERENCES

[1] WU Li-xin; SHI Wen-zhong. Principles and Algorithms of Geographical Information System. Beijing: Science Press, 2003, P:315.(in Chinese)

[2] LIU Jian-yong. Informatics of War Field Environment. Nanjing: Institute of Engineering, PLA University of Science and Technology, 2002.(in Chinese)

[3] C. Murphy; G. Kaiser; and M. Arias. An approach to software testing of machine learning applications. In Proc. of the 19th international conference on software engineering and knowledge engineering, 2007, P:167-172.

[4] FENG Jian-qiang. On the effectiveness of metamorphic testing for numerical programs. The HKU Scholars Hub, 2004, P:8-12.

[5] Y. Chen; S. C. Cheung; and S. Yiu. Metamorphic testing: a new approach for generating next test cases. Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, 1998.

[6] MapInfo Corporation. MapX reference guide. Troy NY, 1999, P:229.

[7] C. Murphy, G. Kaiser, L. Hu, and L. Wu. Properties of machine learning applications for use in metamorphic testing. In Proc. of the 20th International Conference on Software Engineering and Knowledge Engineering, 2008, P:867–872.

Published by Atlantis Press, Paris, France.