

A Combination of Clustering Techniques and Fuzzy Control in 2D Polygon Determination for the Terrain Splitting and Mapping Problem

Le Hoang Son, Nguyen Dinh Hoa, Pier Luca Lanzi, and Bui Thi Huong Lan

Abstract—Recently, the Terrain Splitting and Mapping problem [11] has been considered as one of some current striking trends in GIS researches. This problem is very important in making 3D WebGIS systems and handling large terrain data. In fact, these have been some algorithms dealing with it based on Conditional partitioning approaches [11], Greedy strategies [12], and Heuristic Optimization [17]. However, these algorithms only count for major parameters when minimizing the memory space saving percents such as the error parameter between blocks, the number of processors, and the number of polygons in 2PVD. A minor parameter that can bring much effect to outputted results is the shapes of polygons in 2PVD. Depending on which shapes of polygons are, the answers of whether a solution may exist for this terrain or which algorithm is suitable can be quickly determined. In this paper, we will present a method to determine the shapes of polygons in 2PVD by combining clustering techniques and fuzzy control. The final result of our method is a value showing which algorithm in the literatures [11], [12], [17] is the most suitable for the inputted map. Some experiments in the last section have shown the efficiency of our method for various kinds of polygons in 2PVD.

Index Terms—Clustering, Fuzzy control, Polygon determination, Terrain Splitting and Mapping.

I. INTRODUCTION

3D Geographical Information Systems on Web environment (WebGIS) [2], [5], [15], [16], [18], [19], [22], [23] are regarded as useful tools to model, visualize, and perform analysis operations in 3D terrains. Normally, these terrains are often in grid DEM standards [4], [6], [13], [14], [20], [21], [24] containing matrices of elevation values of some major points. These values form the surface of terrain and are often varied depending on their resolution. Therefore, a large terrain will bring about more elevation values. Hence, the size of terrain is increased as a result. This may prevent the terrain displaying and attribute information querying in Web environment. Indeed, this limitation is really a major obstacle to deploy high performance 3D WebGIS applications as well as advance analysis functions.

This difficulty was stated in the Terrain Splitting and Mapping (TSM) problem [11]. This problem is very important in making 3D WebGIS systems and handling large terrain data. Besides, it was considered as one of some current striking trends in GIS researches [7], [8], [9].

Manuscript received April 30, 2011; revised June 24, 2011. This work is supported by a research grant of Vietnam National University, Hanoi for promoting Science and Technology.

Le Hoang Son is the corresponding author (e-mail: sonlh@vnu.edu.vn).

In the TSM problem, 2D Polygonal Vector Data (2PVD) are provided in equivalent to a specific DEM terrain. They are a set of 2D polygons which represent some special subjects such as Rivers, Lakes, Towns, Land use, etc. Yet they still have common relation to the original DEM terrain and form a 2D-3D GIS dataset. An implementation of the TSM problem [11] in real applications can be found in the literature [10] in which users can choose a specific 2D polygonal layer and view its small, related 3D terrain (Fig 1, Fig 2).

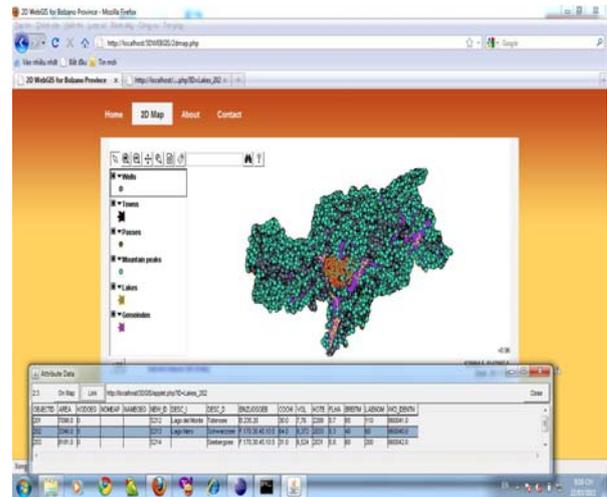


Fig. 1. Query a 2D polygonal layer

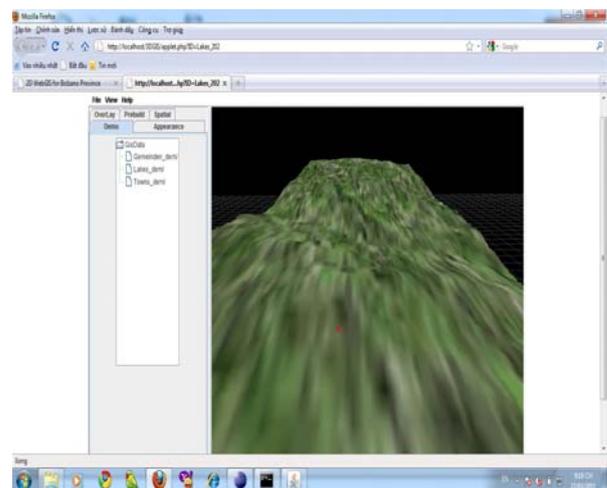


Fig. 2. Link a 2D layer to related 3D terrains

The authors in [11] have proposed SESA algorithm to split the original DEM terrain into some small ones related to each 2D polygon layer in 2PVD for the benefit of saving memory in each processor. Therefore, we can easily view the small terrain through Web interfaces. Besides, a visual link

between 2D attribute data and this terrain is set up. Thus, we can query attribute information from the 3D terrain.

Although the SESA algorithm solved the TSM problem, however, its running time is quite low. Then, two novel algorithms based on Greedy strategy [12] and Heuristic Optimization [17] have been presented to reach that objective in acceptable running time. In fact, these algorithms were tested through experiments and proved to outperform than SESA algorithms in many cases.

However, in those experiments, they only count for major parameters such as the error parameter between blocks, the number of processors, and the number of polygons in 2PVD. These values are the most important parameters but they are not enough to reflect inputted maps and the quality of algorithms [11], [12], [17]. Depending on the shapes of polygons in 2PVD, these algorithms start to divide the original DEM terrain into some blocks. In case of bad polygons (Fig 3), not only is the size of each block not reduced, the running time is increased either.

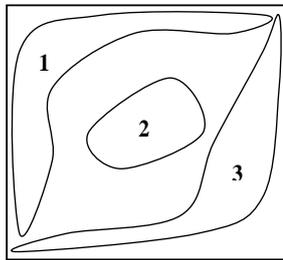


Fig. 3. Bad polygons [11]

In this paper, we will present a method to determine the shapes of polygons in 2PVD by combining clustering techniques and fuzzy control. Then, an outputted value will show which algorithm in the literatures [11], [12], [17] will be suitable for this kind of terrain. In case of bad polygons, for example in Fig 3, this method will quickly conclude without running the whole algorithm as before. Thus, we can assume that the proposed method in this paper is a pre-processing step for the main algorithms in [11], [12], [17].

The rest of this paper is organized as follows. Section 2 presents a classification of polygons in 2PVD. An algorithm to specify preliminary thresholds will be presented in Section 3. Section 4 presents general method combining fuzzy control to specify the shapes of polygons in 2PVD. A numerical example is shown in Section 5. Section 6 presents some experiments. Finally, conclusions and future works are described in the last section.

II. CLASSIFICATION OF POLYGONS IN 2PVD

In 2PVD, we only consider polygonal data. These data reflect some characteristics of 2D maps and are represented in polygonal forms. Throughout a vast collection of GIS data from different sources such as Bolzano-Bolzen province, Italy (2005), Canada (2000), Chameleon data in Africa (2003), Tam Dao National Park, Viet Nam (2000), etc*, together with GIS specialists' opinions, we have derived

some criteria for the classification as follows.

- ❖ The number of polygons in a 2D map (N).
- ❖ Areas of polygons.
- ❖ Distances between these polygons or regions in this map. These figures show the adjacent level ($AdjLevel$).
- ❖ Density of polygons in a region ($DenIndex$).

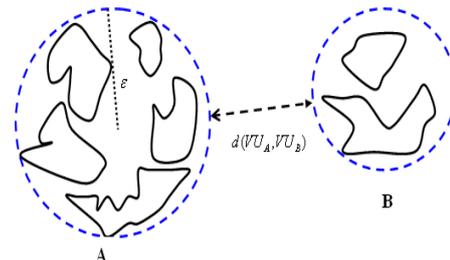


Fig. 4. Two regions in a 2D map

In general, 2D maps are different from each other because of these criteria. For easy implementation, some terms are stated as below.

Def 1: The area of a polygon in 2PVD is calculated as the area of the smallest rectangle containing that polygon. Similarly, the area of a 2D map is equal to the area of smallest rectangle consisting of all polygons in this map.

Roughly speaking, this convention is designed for the suitability with grid DEM terrains as stated in [11].

Def 2: The number of polygons whose areas are equal to the area of original 2D map is denoted as N_c

$$N_c = \left| \left\{ U_i / S(U_i) = S_0 \right\} \right| \quad (1)$$

$$N_c \in [0, N]$$

where U_i is the i^{th} polygon in the 2D Map, $S(U_i)$ is the area of polygon U_i , and S_0 is the area of the 2D map.

Def 3: The number of polygons whose areas are greater than or equal to a threshold $MinPt$ is denoted as N_t

$$N_t = \left| \left\{ U_i / S(U_i) \geq MinPt \right\} \right| \quad (2)$$

$$N_t \in [0, N]$$

Def 4: Assume that the number of regions in the 2D map after clustering all polygons is m_v . We call $N_D \in [0, m_v]$ is the number of regions whose densities are greater than or equal to a threshold $MinDen$.

$$N_D = \left| \left\{ VU_j / DenIndex(VU_j) \geq MinDen \right\} \right| \quad (3)$$

where VU_j is the j^{th} region in the 2D map, $j = \overline{1, m_v}$, $DenIndex(VU_j)$ is the density of region VU_j .

Def 5: The number of pairs of regions whose adjacent levels are smaller than or equal to a threshold $MaxAdj$ is attributed to $N_A \in [0, C_{m_v}^2]$.

$$N_A = \left| \left\{ \left\{ VU_i, VU_j \right\} / AdjLevel(VU_i, VU_j) \leq MaxAdj \right\} \right| \quad (4)$$

where $AdjLevel(VU_i, VU_j)$ is the adjacent level of two

* These data are available at this address
<http://www.megaupload.com/?d=CMYBJG18>

regions VU_i and VU_j , $i = \overline{1, m_v}$, $j = \overline{1, m_v}$, $i \neq j$.

Basically, we can distinguish the shapes of polygons in 2PVD through five parameters above. Some examples can be extracted from the GIS dataset above.

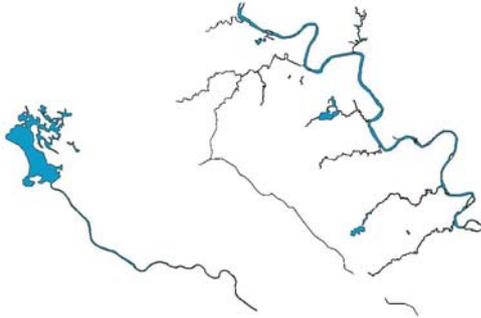


Fig. 5. Rivers



Fig. 6. Thai Nguyen, Viet Nam administrative map

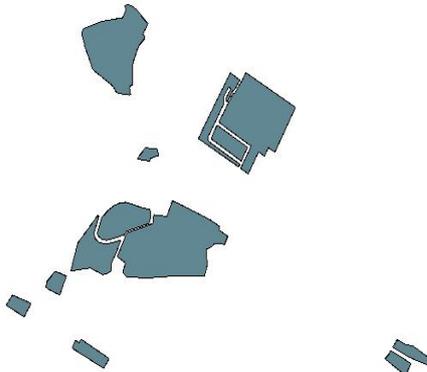


Fig. 7. Factories map

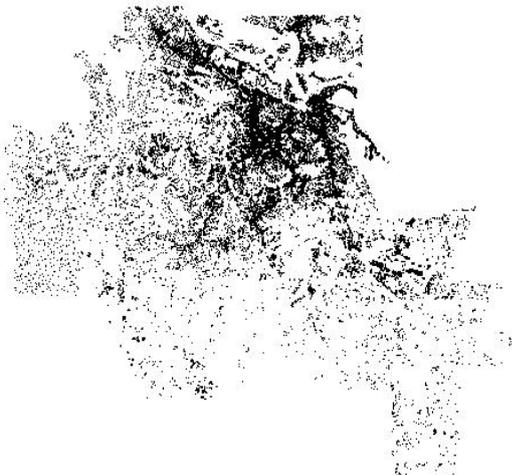


Fig. 8. Household distributive map

Values of parameters in each figure above are

$$\diamond N = 84, N_c = 0, N_t \in \left[\frac{N}{2}, \frac{3N}{4} \right], N_D \in \left[\frac{m_v}{2}, \frac{3m_v}{4} \right], \text{ and } N_A < \frac{C_{m_v}^2}{2} \text{ (Fig 5).}$$

$$\diamond N = 27, N_c = 0, N_t \geq \frac{3N}{4}, N_D \geq \frac{3m_v}{4}, \text{ and } N_A \geq \frac{3C_{m_v}^2}{4} \text{ (Fig 6).}$$

$$\diamond N = 13, N_c = 0, N_t \geq \frac{3N}{4}, N_D \geq \frac{3m_v}{4}, \text{ and } N_A \in \left[\frac{C_{m_v}^2}{2}, \frac{3C_{m_v}^2}{4} \right] \text{ (Fig 7).}$$

$$\diamond N = 34231, N_c = 0, N_t \in \left[\frac{N}{2}, \frac{3N}{4} \right], N_D \leq \frac{m_v}{2}, \text{ and } N_A < \frac{C_{m_v}^2}{2} \text{ (Fig 8)}$$

Consequently, we will use these parameters for some algorithms in next sections.

III. A CRISP ALGORITHM

The first task of 2D polygonal determination is to bring out a preliminary decision of the ‘right’ shape of polygons. It is considered as the first filter of which shape the inputted map can be. Moreover, this algorithm provides an orientation for the main algorithm which will be presented in the next section. In fact, the whole idea is based on experts’ experiences of some possible cases of 2D maps and is expressed by the scale of shapes (Fig 9). This scale shows various forms of maps and can be used further to determine the most suitable algorithm from [11], [12], [17].

The crisp algorithm contains the following steps.

Crisp ($N, N_0, MinPt, \xi, MP, MinDen, MaxAdj$)

1. Calculate the areas of polygons ($S(U_i), i = \overline{1, N}$) and the area of 2D map (S_0).
2. Determine the value of N_c and N_t following by the areas above and parameter $MinPt$.
3. Use DBSCAN algorithm [3] to cluster all polygons in 2D map into m_v regions with each region’s radius is ξ and the minimal number of points required to form a region is MP .
4. Find the maximal number of points in all regions $MaxDen$.
5. Update the $MinDen$

$$MinDen = Rand(MP, MaxDen) \quad . (5)$$

6. Determine the value of N_D following by the new threshold $MinDen$.
7. Calculate the matrix of distances between regions. Then, compare these distances with a threshold $MaxAdj$ and determine the value of N_A .
8. Calculate the value EI following by $N, N_c, N_0, N_t, N_D, m_v$ and N_A .

$$EI = F_{NC} \left(is \left(\frac{N}{N_0} \right) + \frac{N_t}{N} + \frac{N_D}{m_v} + \frac{N_A}{C_{m_v}^2} \right) \quad . (6)$$

where

$$F_{Nc}(x) = \begin{cases} x & N_c > 0 \\ -1 & \text{otherwise} \end{cases} \quad (7)$$

$$is\left(\frac{x}{y}\right) = \begin{cases} 1 & x \geq y \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

9. Based on the scale of shapes below (Fig 9), calculate the membership value of this 2D map.

$$P_i = 1 - \frac{d(EI, v_i)}{\sum_{j=1}^l d(EI, v_j)} \quad (9)$$

where l is the number of segments in the scale of shapes covering the value EI . v_i is the center of i^{th} segment and $d(EI, v_i)$ is the distance between EI and v_i .

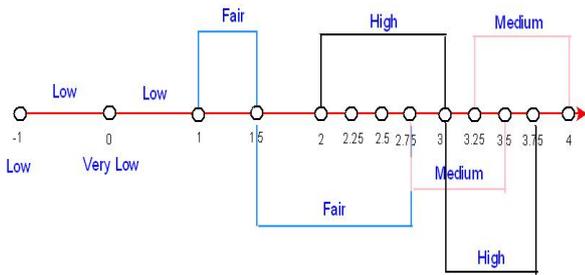


Fig. 9. A crisp scale of shapes

10. Conclude the possible shapes of polygons in the 2D map with membership values are calculated above.

IV. 2D POLYGON DETERMINATION ALGORITHM

Although we can use the preliminary result from the crisp algorithm above for the selection of algorithm, however, it still remains some disadvantages. First, if the inputted map belongs to many shapes then we can only choose the ‘right’ shape through its membership value. Therefore, a misleading selection can be happened with the probability calculated as complement of the membership value. Second, some markers in the scale of shapes are extracted from experiences of GIS experts. Somehow, they can not be approved by other experts. In the other words, we need a flexible understanding of “High”, “Medium”, etc. maps.

These two shortcomings lead us to think about *Fuzzy Control Systems* (FCS). A FCS is a control system based on fuzzy logic- a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 0 or 1 (true or false). Fuzzy controllers are very simple conceptually. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels, and so on, to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value. Details of the mechanism of FCS can be found at Ref [1].

The figure below shows the mechanism the *2D Polygon Determination Algorithm* (2PDA) based on Fuzzy Control Systems (Fig 10).

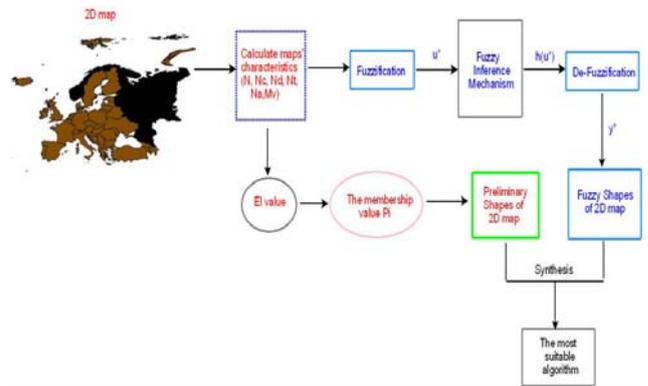


Fig. 10. Mechanism of 2PDA

Summarization of the mechanism is shown below.

Step 1: From the inputted 2D map, calculate the maps’ characteristic through parameters $N, N_c, N_0, N_t, N_D, m_v$ and N_A .

Step 2: Perform the crisp algorithm in previous section to find the preliminary shapes of 2D map.

Step 3: Send these parameters to Fuzzy Controller and calculate a fuzzy value representing for the inputted 2D map.

3.1. Fuzzy Controller is designed as five inputs and one outputs.

Input: $N, N_c = \{high, low\}$

$N_t, N_D, N_A = \{high, medium, low\}$

Output: $map = \{high, fair, medium, low, verylow\}$

3.2. The membership function of Input and Output variables are the trapezoid function.

3.3. The inference mechanism is decided by fuzzy rules. In our problem, these are 108 rules which determine the suitable map for a given inputted map. Some rules, for example, are

R_1 : If N is high and N_c is low and N_t is high and N_D is high and N_A is high then map is medium.

R_2 : If N is high and N_c is low and N_t is medium and N_D is medium and N_A is low then map is high.

R_3 : If N is high and N_c is low and N_t is high and N_D is high and N_A is medium then map is high.

R_4 : If N is high and N_c is low and N_t is medium and N_D is low and N_A is low then map is fair.

R_5 : If N is high and N_c is low and N_t is medium and N_D is high and N_A is medium then map is medium.

R_6 : If N is high and N_c is low and N_t is low and N_D is low and N_A is low then map is fair.

R_7 : If N is high and N_c is high and N_t is medium and N_D is low and N_A is medium then map is low.

R_{8*} : If N is low then map is very low.

Other fuzzy rules return the value “low” of outputted variables.

Rule R_{8*} represents for 54 rules in the system. Regardless of values of the remainders, these rules return “map is very

low” if the value of N is “low”.

3.4. The Deductive and Composition methods are Sum-Min.

3.5. De-fuzzification: Use the centre of gravity method from Ref [1].

Step 4: Synthesize the Fuzzy Shape of the 2D map and its preliminary results to form a list of possible solutions. The priority is for the Fuzzy shape.

Step 5: Decode the shape of inputted 2D map and choose the appropriate algorithm

If map is high then use the PSO-TSA method [17].

If map is fair then use the GA-TSA method [17].

If map is medium then use the R-SESA method [12].

If map is low then use the SESA method [11].

If map is very low then do nothing and conclude immediately that no solution is found.

Obviously, we can totally check a suitable algorithm through the 2PDA method for a given inputted 2D map related to a DEM terrain. In case of no solution, this method also quickly answers without running the whole algorithm in [11], [12], [17].

V. A NUMERICAL EXAMPLE

In this part, we will illustrate the 2PDA algorithm by a numerical example. Let us see the Fig 11.



Fig. 11. A 2D map

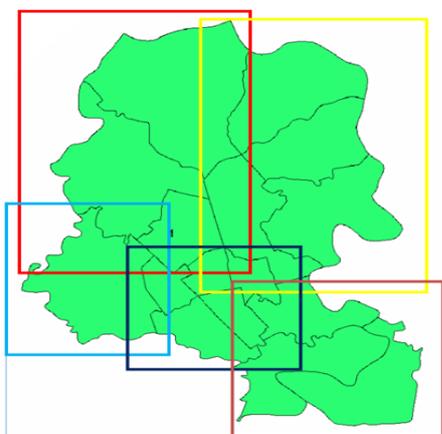


Fig.12. The map after clustering

Using DBSCAN method, this map is divided into 5 regions (Fig 12). Through the crisp algorithm, we easily calculate these following parameters

- ❖ The number of polygons $N = 16$
- ❖ The polygon’s threshold $N_0 = 3$
- ❖ $N_c = 0$
- ❖ $N_t = 9$
- ❖ $N_D = 3$
- ❖ $m_v = 5$
- ❖ $N_A = 7$

Then, the value of EI is

$$EI = 1 + \frac{9}{16} + \frac{3}{5} + \frac{7}{10} = 2.8625$$

From the scale of shapes (Fig 9), these are two kinds of this map

The map is high (a)

or

The map is medium (b).

The membership value of the map to the first shape (a) is

$$P_1 = 1 - \frac{2.8625 - 2.5}{(2.8625 - 2.5) + (3.125 - 2.5)} = 63.3\%$$

Similarly, the membership value of the map to the second shape (b) is

$$P_2 = 36.7\%$$

Therefore, the preliminary decision of the ‘right’ shape of the map is ‘high’ with the membership value is 63.3%.

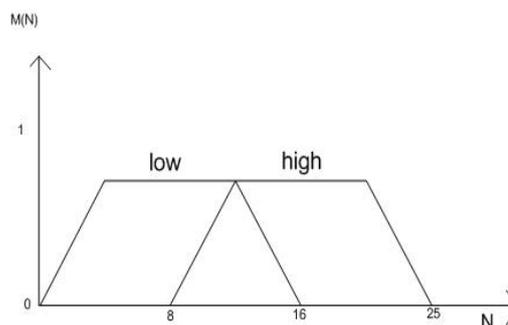


Fig. 13. The membership function of variable N

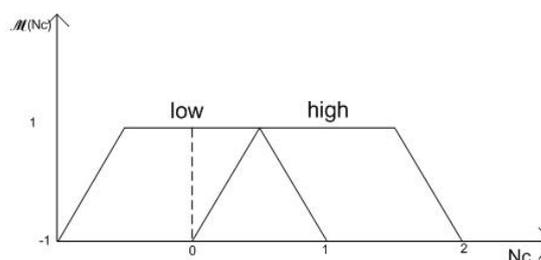


Fig. 14. The membership function of variable N_c

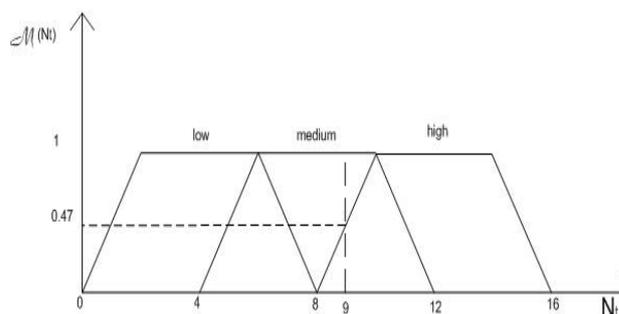


Fig. 15. The membership function of variable N_t

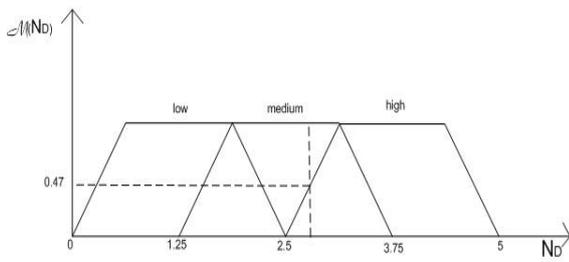


Fig. 16. The membership function of variable N_D

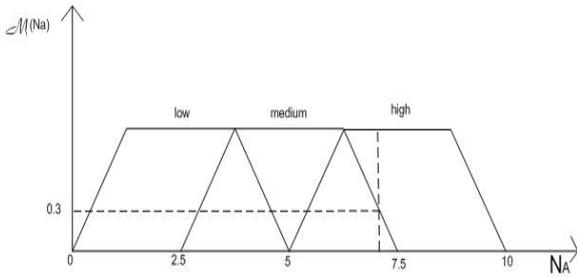


Fig. 17. The membership function of variable N_A

Next, we use Fuzzy Controller to check this decision. The membership function of Input and Output variables are shown in the figures (Fig 13, Fig 14, Fig 15, Fig 16, Fig 17).

Use the Deductive method Min to calculate the membership value of each rule

$$\begin{aligned} \mu_{R1} &= \text{Min}\{\mu_{high}(x_1), \mu_{low}(x_2), \mu_{high}(x_3) \\ &\quad \mu_{high}(x_4), \mu_{high}(x_5), \mu_{medium}(y_1)\} \\ &= \text{Min}\{1; 1; 0.47; 0.47; 1; \mu_{medium}(y_1)\} \\ &= \text{Min}\{0.47; \mu_{medium}(y_1)\} \\ \mu_{R2} &= \text{Min}\{1; 1; 1; 0; \mu_{high}(y_2)\} = 0 \\ \mu_{R3} &= \text{Min}\{1; 1; 0.47; 0.47; 0.3; \mu_{high}(y_3)\} \\ &= \text{Min}\{0.3; \mu_{high}(y_3)\} \end{aligned}$$

$$\begin{aligned} \mu_{R4} &= \text{Min}\{1; 1; 1; 0; 0; \mu_{fair}(y_4)\} = 0 \\ \mu_{R5} &= \text{Min}\{1; 1; 1; 0.47; 0.3; \mu_{medium}(y_5)\} \\ &= \text{Min}\{0.3; \mu_{medium}(y_5)\} \\ \mu_{R6} &= \text{Min}\{1; 1; 0; 0; 0; \mu_{fair}(y_6)\} = 0 \\ \mu_{R7} &= \text{Min}\{1; 0; 1; 0; 0.3; \mu_{low}(y_7)\} = 0 \\ \mu_{R8^*} &= 0 \\ \mu_{R_other} &= 0. \end{aligned}$$

Results of the Composition method or Sum-Min device are shown below.

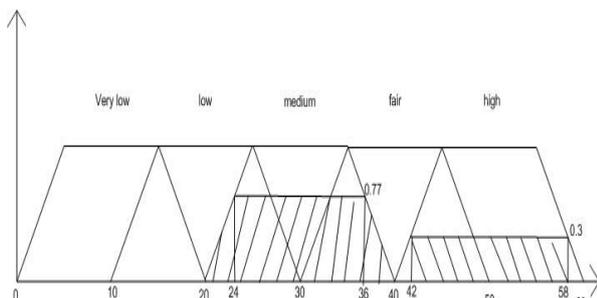


Fig. 18. Composition method

Then, we use the centre of gravity method [1] for de-fuzzification.

$$y^* = \frac{\int_S y \mu_R(y) dy}{\int_S \mu_R(y) dy} = \frac{\sum_{i=1}^n \left(\int_S y \mu_{R_i}(y) dy \right)}{\sum_{i=1}^n \left(\int_S \mu_{R_i}(y) dy \right)}$$

$$y^* = \frac{369.6 + 470.25}{12.32 + 8.5} = 40.33$$

Following by the Fig 18, we can conclude that “the map is high” and the suitable algorithm is PSO-TSA [17].

VI. EXPERIMENTS

In this section, we have implemented the proposed algorithms 2PDA in C programming language and executed them on a PC Intel Quad Core, 2.1 GHz, 2GB RAM. Some parameters of the algorithm were initialized as below.

- ❖ $N_0 = 3$
- ❖ $MinPt = S_{min} + 0.25 \times (S_{max} - S_{min})$ where S_{max} and S_{min} are the maximal and minimal areas of polygons in the inputted dataset, respectively.
- ❖ $\xi = D_{min} + 0.05 \times (D_{max} - D_{min})$ where D_{max} and D_{min} are the maximal and minimal distances between polygons, respectively.
- ❖ $MP = 2$
- ❖ $MinDen = MP + 0.25 \times (MaxDen - MP)$ where $MaxDen$ is described in the Crisp algorithm.
- ❖ $MaxAdj = d_{min} + 0.5 \times (d_{max} - d_{min})$ where d_{max} and d_{min} are the maximal and minimal distances between regions after using DBSCAN algorithm.

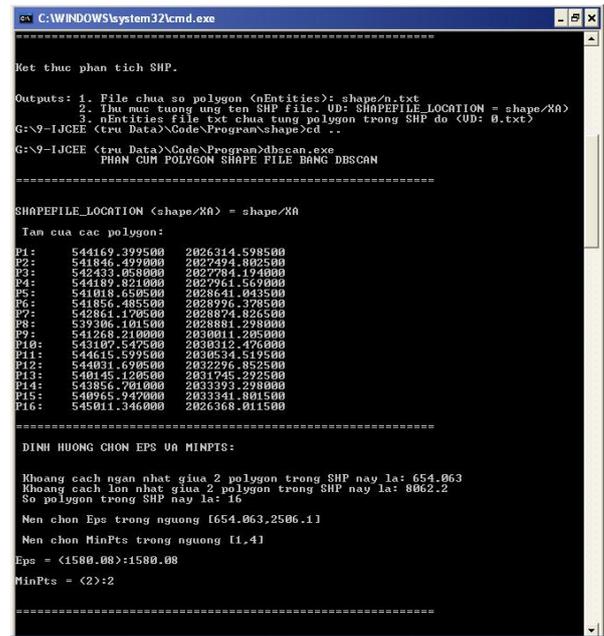


Fig. 19. Implementation of 2PDA

First, we study the change of the value EI and the value of outputted variable in Fuzzy Controller following by

different number of polygons in equivalent to different 2D maps and some parameters above. Test data are taken from GIS data in Section 2 and are classified into two groups: Training set and Tested set. The training dataset is used to train the 2PDA algorithm in order to extract the suitable parameters such as $N_0, MinPt, \zeta, MP, MinDen$ and $MaxAdj$ as stated at the beginning of this section. The other one is used for these tests below.

TABLE I: THE CHANGES OF EI AND y^*

N	16	27	61	1068
N_0	3	3	3	3
$MinPt$	4812346	38957083	9982628984	74205707946
ξ	1024	4286	99391	229299
MP	2	2	2	2
$MaxDen$	3	8	26	820
$MinDen$	2.25	3.5	8	206.5
$MaxAdj$	4358	24440	717398	2262159
EI	2.33	2.6	2.38	1.914
y^*	26.25	27.5	29.8	29.9

In Table 1, we calculate the value of EI and y^* through some 2D maps containing different number of polygons and shapes that are expressed by different parameters. For example, in the first case, the value of EI in the scale of shapes (Fig 9) shows that the type of this map are Fair and High with membership values 0.8 and 0.2, respectively. Then, this map is inputted into Fuzzy Controller to filter the results. The value y^* shows that its real shape is Fair. Similarly, the results of third case are (Fair, High) with membership values (0.9; 0.1) from the crisp algorithm and real shape is Fair.

However, it is possible that shapes with small membership values in the scale of shapes sometime turn out to be the real shapes. For instance, in second case, the value $EI = 2.6$ means (Fair, High) with membership values (0.32; 0.67). But the value $y^* = 27.5$ confirms that real shape of the inputted 2D map is Fair. This shows the necessity of fuzzy control in finding the correct answer.

The last case in our test shows the unity of two results from crisp algorithm and Fuzzy control where both of them return Fair value. In these cases, we do not need to perform the Fuzzy step in general because results are obvious. In addition, this helps us to reduce the computational time.

A remark can be extracted from this test is that the common shape of polygon in most of 2D maps is Fair. Somehow, it can be used as the pre-knowledge to choose correct shapes of polygons and algorithms.

In what follows, we study the distribution of methods from the literatures [11], [12], [17] that are used for various inputted maps from the tested dataset (Fig 20). Thence, we can recognize that the most using method is SESA with 45 percents. The next one is GA-TSA with 33 percents and two last ones in descending order are PSO-TSA and R-SESA with the same 11 percents. From this result, the most common shapes of polygons in 2D maps seem to be Fair and

Low. Therefore, in general, we can predict the shape of polygons in any 2D maps with a high accuracy.

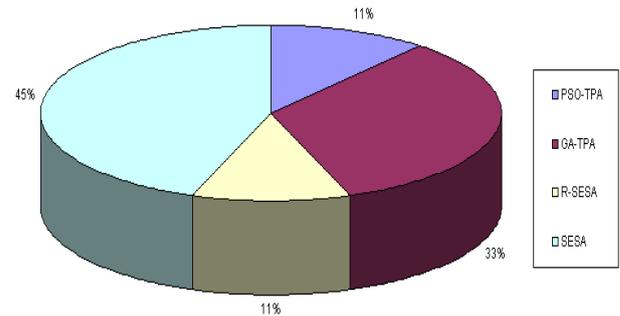


Fig. 20. Distribution of methods

VII. CONCLUSION AND FUTURE WORKS

In this paper, we focus on the 2D Polygon determination in the Terrain Splitting and Mapping problem. Throughout a brief introduction, a great demand of polygon's shape understanding is required before running the main algorithms in the literatures [11], [12], [17] for the benefit of time saving and quickly response. Then, the novel algorithm 2PDA is made for this task. This algorithm contains two basic steps: a crisp processing and fuzzy control mechanism. Results of this algorithm are the shape of polygons for a given inputted map and the most suitable algorithm from the Ref. [11], [12], [17]. Besides, we have shown a numerical example of this algorithm for better understanding as well as performed some experiments to look for the common shapes of polygons and methods. Therefore, the algorithm 2PDA is supposed to be suitable for the pre-processing step in the TSM problem.

In the future, we will look for some multi-objective optimization problems in 3D WebGIS. Moreover, some other stochastic optimization algorithms for the TSM problem are also our targets.

ACKNOWLEDGMENT

The authors wish to thank Prof. Pham Van Cu and Ms. Tran Thi Kim Dung, VNU for providing GIS data. Besides, we are greatly indebted to Mr. Truong Chi Cuong who spent a lot of time to help us complete the source code.

The last thanks will be sent to anonymous reviewers, Prof. Bui Cong Cuong, VAST and the research group at the Center for High Performance Computing, VNU for supporting and giving useful comments.

REFERENCES

- [1] Bui Cong Cuong, Nguyen Doan Phuoc, *Fuzzy Systems, Neural Network and applications* (in Vietnamese), Second Edition, Science and Technology Publishing House, 2006.
- [2] CUI YongyiZHOU, DongruWAN, GangFU Huasheng, "Web VRGIS Study Based on VRML", *Computer Engineering*, vol. 8, 2002.
- [3] Ester M., Kriegel H.-P., Sander J., Xu X, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", In *Proceeding of 2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland, OR, AAAI Press, 1996, pp. 226-231.
- [4] GAO Ying-jie et al, "Development of DEM on 1:10000 Scale and Its Application in Geo-sciences", *Journal of Anhui Agricultural Sciences*, vol. 2, 2009.

[5] LI Lin-lin, CAO Kai-bin, GUAN Bin, ZHU Wei-dong, "Study on the Application of WebGIS and VR in the Information System of Real Estate", *Sci-Tech Information Development & Economy*, vol. 5, 2007.

[6] LIU Yongqiong, WU Yanlan, HU Hai, HU Peng, "Assessment method of digital elevation models accuracy and its limitations", *Journal of Geomatics*, vol. 5, 2009.

[7] Le Hoang Son, "On the Development of Three Dimensional WebGIS Systems: Some New Trends and Prospects", In *Proceedings of the 2010 3rd IEEE International Conference on Computer Science and Information Technology* (IEEE ICCSIT 2010), July 9 - 11, 2010, Chengdu, China, vol. 1, pp. 182 - 186.

[8] Le Hoang Son, "An Approach to Construct SGIS-3D: a Three Dimensional WebGIS System Based on DEM, GeoVRML and Spatial Analysis operations", In *Proceedings of the 2nd IADIS International Conference Web Virtual Reality and Three-Dimensional Worlds 2010* (IADIS Web3DW 2010), July 27 - 29, 2010, Freiburg, Germany, pp. 317 - 326.

[9] Le Hoang Son, "An Exploratory Study about Spatial Analysis Techniques in Three Dimensional Maps for SGIS-3D systems", In *Proceedings of the 2010 IEEE International Conference on Electronics and Information Engineering* (IEEE ICEIE 2010), August 1 - 3, 2010, Kyoto, Japan, vol. 1, pp. 199 - 203 (ISI Proceeding).

[10] Le Hoang Son, Pham Huy Thong, Nguyen Duy Linh, Truong Chi Cuong and Nguyen Dinh Hoa, "Developing JSG Framework and Applications in COMGIS Project", *International Journal of Computer Information Systems and Industrial Management Applications* (IJCSIM), vol. 3, 2011, pp. 108 - 118.

[11] Le Hoang Son, Pham Huy Thong, Nguyen Duy Linh, Nguyen Dinh Hoa, and Truong Chi Cuong, "Some Results of 3D Terrain Splitting By 2D Polygonal Vector Data", *International Journal of Machine Learning and Computing* (IJMLC), vol. 1, no. 4, October 2011 (accepted).

[12] Le Hoang Son, Pham Huy Thong, Truong Thi Hanh Phuc, Nguyen Dinh Hoa, Nguyen Thi Hong Minh, "Some Extensions of Terrain Splitting and Mapping Problem", *International Journal of Computer Theory and Engineering* (IJCTE) (accepted).

[13] M. van Kreveld, *Digital Elevation Models: overview and selected TIN algorithms*, Algorithmic Foundation of GIS, Springer-Verlag, 1997.

[14] MA Chi, SONG Wei-dong, "Creating urban DEM by use of topographic map with DWG format", *Journal of Anshan University of Science and Technology*, vol. 5, 2004.

[15] Miao Xuelan, "Application of vrm and webgis technology to the traveling geographic information system", *Computer Applications and Software*, vol. 7, 2005.

[16] MA Pengfei, ZHAO Wenji, HU Zhuowei, DUAN Fuzhou, CAI Wenbo, "Research of 3-Dimensional Visualization of Rural Folk Tourism Sight", *Geospatial Information*, vol. 5, 2009.

[17] Nguyen Duc Thien, Le Hoang Son, Pier Luca Lanzi, and Pham Huy Thong, "Heuristic Optimization Algorithms For Terrain Splitting and Mapping Problem", *International Journal of Engineering and Technology* (IJET) (accepted).

[18] SONG Wei, LI Hua, "Research of 3D Web GIS system based on X3D", *Computer Engineering and Design*, vol. 11, 2005.

[19] SHI Rong, XU Hui-ping, Chen Hua-gen, "Application of 3D Virtual WebGIS in Earthquake Disaster Prevention and Mitigation", *Journal of Seismological Research*, vol. 2, 2008.

[20] Sandra Lanig, Arne Schilling, Beate Stollberg and Alexander Zipf, "Towards Standards-Based Processing of Digital Elevation Models for Grid Computing through Web Processing Service (WPS)", In:

Computational Science and Its Applications (ICCSA 2008), Lecture Notes in Computer Science, 2008, Volume 5073/2008, pp. 191-203.

[21] S. Rayburga, M. Thomsa and M. Neave, "A comparison of digital elevation models generated from different data sources", *Geomorphology*, vol. 106, issues 3-4, 15 May 2009, pp. 261-270.

[22] WANG Feng, LIU Ren-yi, LIU Nan, "Study on the application of WebGIS and virtual reality technology in tourism development", *Journal of Zhejiang University (Sciences Edition)*, vol. 6, 2005.

[23] WANG Wei, WU Sheng, "Research on the Integration of 3D Geographic Information Web Services", *Geomatics World*, vol. 1, 2008.

[24] WANG Ke-ke, ZHANG Li-chao, PAN Zhen, WANG Qing-shan, ZHANG Shi-quan, "Research on Multiresolution Dynamic Creating Networks Algorithm of DEM based on DirectX", *Beijing Surveying and Mapping*, vol. 2, 2008.



Le Hoang Son is a researcher at the Center for High Performance Computing, Hanoi University of Science, VNU. He is a member of IACSIT and also member of the editorial board of the International Journal of Engineering and Technology (IJET). His major field includes Data Mining, Geographic Information Systems and Parallel Computing. Email: sonlh@vnu.edu.vn



Nguyen Dinh Hoa is an associate professor and vice director of the Information Technology Institute, VNU. His research areas include linear programming, optimization, data structure and algorithms, and Geographic Information Systems. He is member of the organizing committees of many prestigious national conferences since 1998. Email: hoand@vnu.edu.vn



Pier Luca Lanzi was born in Turin, Italy, in 1967. He received the Ph.D. degree in Computer and Automation Engineering from the Politecnico di Milano in 1999. He is associate professor at the Politecnico di Milano, Dept. of Electronics and Information. His research areas include evolutionary computation, reinforcement learning, machine learning. He is interested in applications to data mining and computer games. He is member of the editorial board of the Evolutionary Computation Journal, the IEEE Transaction on Computational Intelligence and AI in Games, and Evolutionary Intelligence. He is also the founding editor in chief of the SIGEVolution, the newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation. He served as the general chair for the 2009 IEEE Conference on Computational Intelligence in Games (CIG-2009) and for the 2011 Genetic and Evolutionary Computation Conference (GECCO-2011). Email: lanzi@elet.polimi.it



Bui Thi Huong Lan is a collaborator at the Center for High Performance Computing, Hanoi University of Science, VNU. Her research interests include Geographic Information Systems and Algorithms. Email: huonglanbt_bk@yahoo.com