The method to determine position in 3D digital sand table

Mac Van Vien*

Military Information Technology Institute, Academy of Military Science and Technology.
*Corresponding author: vienmvit@gmail.com
Received 15 Sep 2022; Revised 5 Dec 2022; Accepted 15 Dec 2022; Published 30 Dec 2022.
DOI: https://doi.org/10.54939/1859-1043.j.mst.CSCE6.2022.16-27

ABSTRACT

This paper proposes a method to determine the working position in the 3D digital sand table. Starting from analyzing the actual business of the 3D digital sand table to propose the operation cases and build the algorithm to determine the position for each of those cases. Then, the presentation of the application of that method to specific functions of the GlobeDesktop system. Finally, the paper presents experimental content to evaluate the method.

Keywords: Sand table; 3D digital sand table; Skyline Terraexplorer Pro; Map; Digital Map.

1. INTRODUCTION

The 3D digital sand table is increasingly widely used in combat, determining the position on the digital table to draw symbols, measure, and determine trajectory, etc. is a regular and very important job. With 2D digital maps, a cursor location corresponds to a point on the map. With a 3D digital sand table, a cursor location corresponding to a straight line in three-dimensional space is depicted as the dashed line in figure 1, so the task of the paper is to find a method to determine obtain the position in space from a point on the screen (the location of the cursor).

From the practical requirements of the digital sand table, and based on the business analysis of the GlobeDesktop system, the article proposes three working cases where it is necessary to determine the position. If there is a need to perform more than one case, it must be separated into each case in turn:

1- Operation on terrain: move the cursor on the screen to get the position on the terrain (P1). This case works to write and draw symbols, measure terrain distances, and determine orbits. Depending on the operational object, there may be a relatively small difference in altitude from the terrain;

2- Operation to determine altitude: move the cursor on the screen to determine the height of the position in space relative to a given position (P2). In this business, case to measure the height, change the height of the symbol from the original height;

3- Operation on fixed altitude: move the cursor on the screen to define a position whose height is equal to the height of a given point (P3). This case of cooperation is used to move a symbol in the air or underground while maintaining the same altitude.

The modeling of problem:

Input: Terrain display on a 3D digital sand table, camera position L_c (including x, y, altitude, yaw, pitch, roll), camera field of view, location of the cursor on screen S(X, Y), and the case of working:

- P1: Altitude difference Δh;
- P2: The mark position $L_1(x_1,y_1,z_1)$;
- P3: The altitude h; and the mark position $L_1(x_1,y_1,h)$.

Output: Working position L(x,y,z).

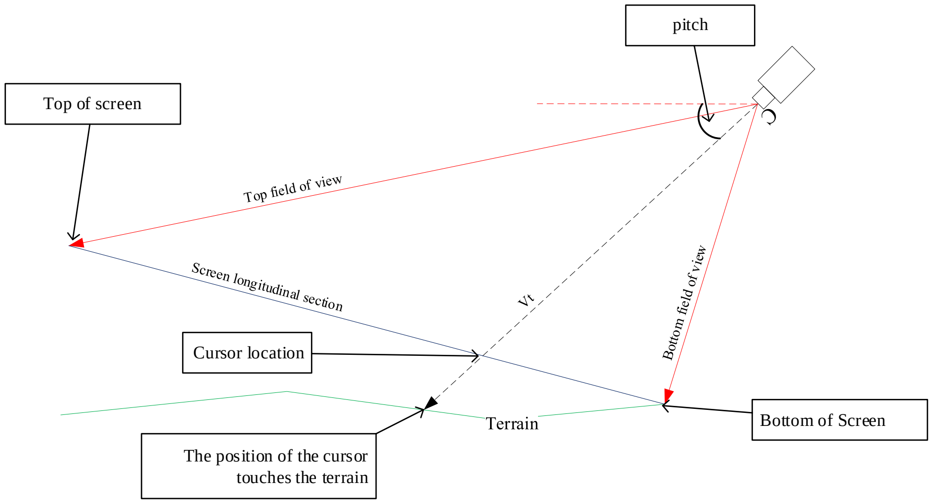


Figure 1. *Longitudinal section of the screen at and locattion of cursor.*

2. PROPOSE A METHOD TO DETERMINE THE POSITION

2.1. Operation on terrain (P1)

Move the cursor to track the terrain, and determine the operational position about Δh from the ground. When the cursor is on the terrain, always determine the point $L_d(x_d, y_d, z_d)$ as the point from the pointer to the terrain. The altitude of the operational position has the value: $z_d + \Delta h$.

$$L = L_c \textbf{\textit{ move }} d_{distance} \textbf{\textit{ to }} L_d;$$

$$\text{Where: } d_{distance} = \left| \frac{zc - (zd + \Delta h)}{\sin(Vc{-}p.Pitch)} \right|$$

The values and distances are described as shown below:

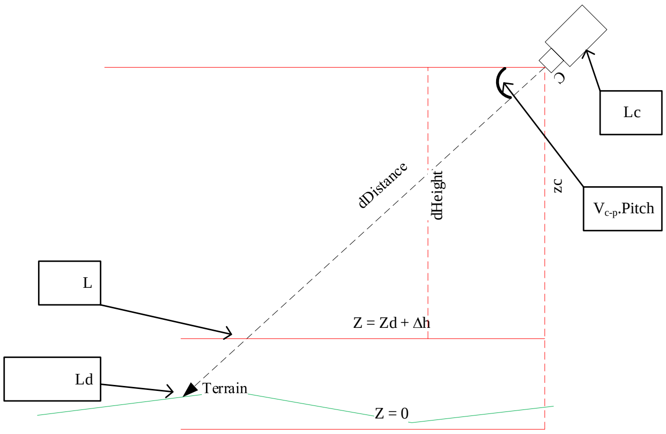
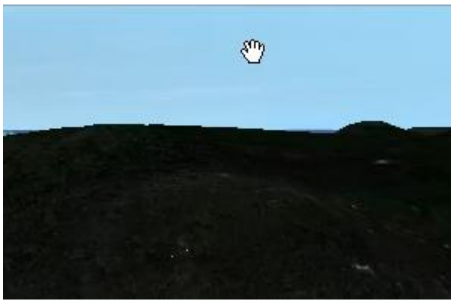


Figure 2. *Illustrating the algorithm to determine the on terrain position.*

2.2. Operation to determine altitude (P2)

Move the cursor to determine the height relative to the mark position $L_1(x_1,y_1,z_1)$. Divide into two cases of cursor location: The cursor "touches" the terrain, and the cursor does not "touch" the terrain.

The cursor "touches" the terrain



The cursor does not "touch" the terrain

Figure 3. *Illustrate two cases of cursor location.*

The cursor "touches" the terrain (P2.1)

The position L_d on terrain is determined to correspond to the cursor point (S). The altitude is determined by the formula: $h = z_c - d_1 * \tan(\mathrm{Vc} - \mathrm{d.Pitch})$

The position is determined to be $L(x_1, y_1, h)$;

The values are described as shown in the figure below.

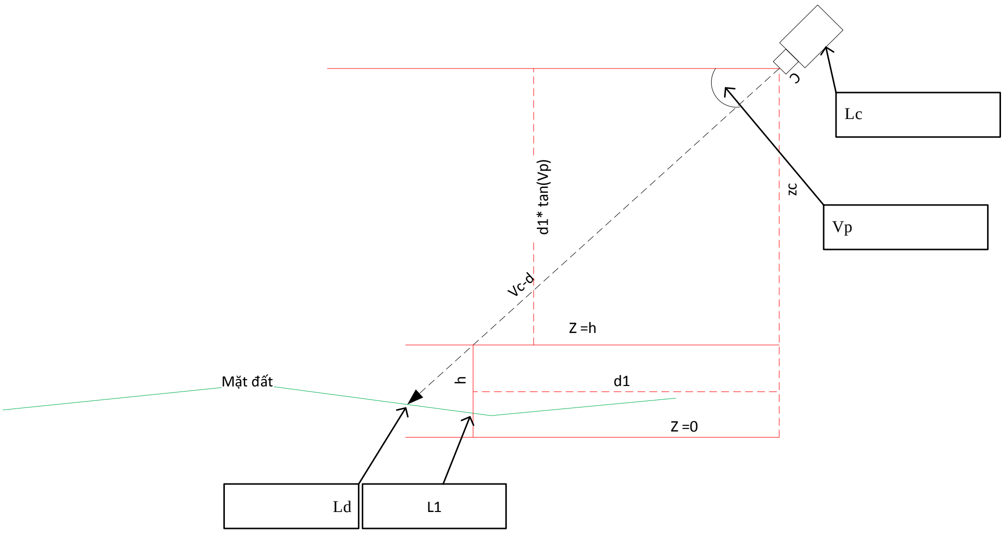


Figure 4. *Illustrate the algorithm to determine the altitude when the cursor "touches" the terrain.*

The cursor does not "touch" the terrain (P2.2)

Use iterations to determine the height difference from the mark position L_1 to the position determined by the cursor location on the screen. To perform the algorithm, it is necessary to determine the height difference value on the screen (pixels) when shifting the altitude on the 3D digital sand table from L_1 to a position determined by the cursor location of 1 meter.

The algorithm consists of two steps:

- Determine initial L position: Calculate ppm (L_1, L_1 move 1 meter to S). Initial altitude is determined by the formula: $L = L_1 + (Y - SL_1.Y) * ppm$;

- Iteration with convergence condition check to correct the position for more accuracy: Perform the iteration by redefining the ppm value from L to L_1. During iteration check the convergence condition, if not, return to the previous position.

The algorithm diagram describes the figure below:

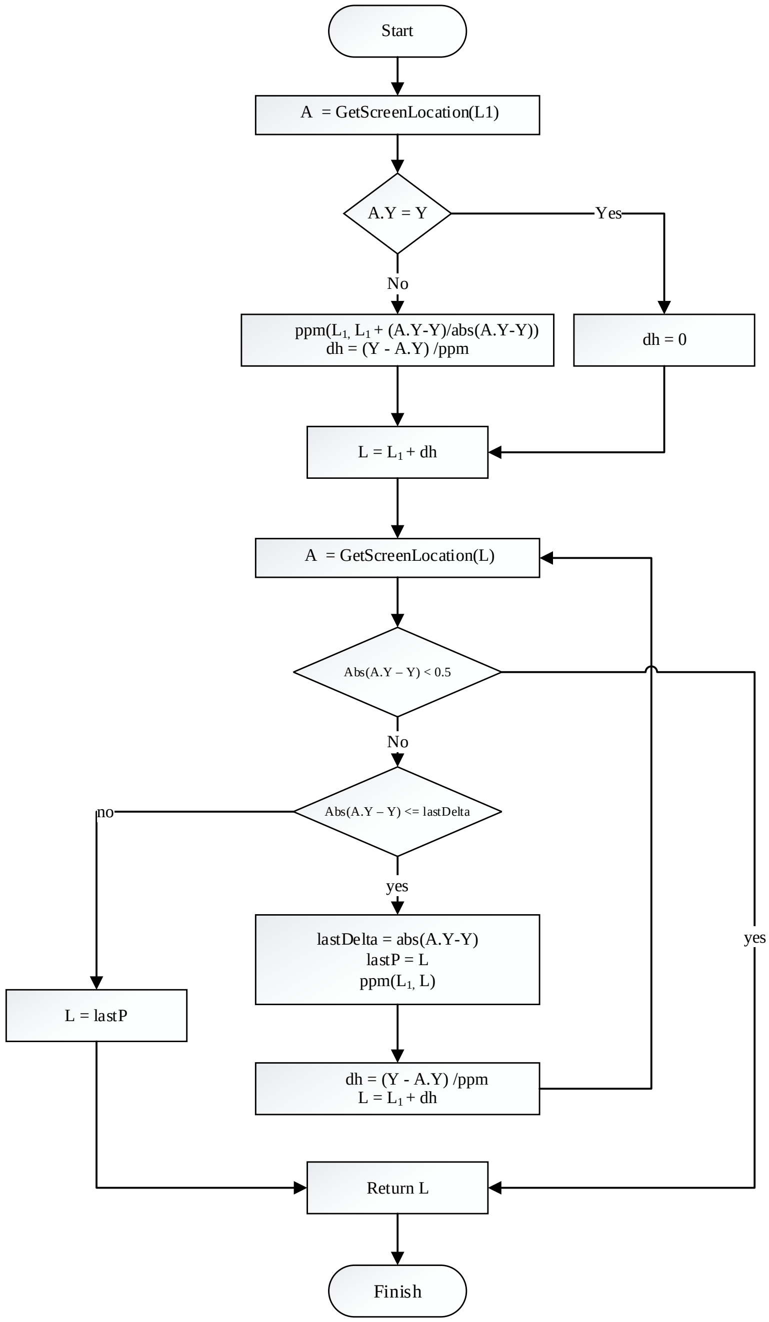


Figure 5. *Algorithm diagram for determining altitude when the cursor does not "touch" the terrain.*

2.3. Operation on fixed altitude (P3)

Move the cursor from a position with a given altitude to another position. The operation is often used in the service of moving the position of the symbol, editing the trajectory of the movement in the sky, etc. In this operation, it is also divided into two cases where the cursor location is: the cursor "touches" the terrain, and the cursor does not "touch" the terrain.

The cursor "touches" the terrain (P3.1)

The algorithm to determine the position is the same as the case mentioned in section 2.2.1. Replace the height value $z_1 + \Delta_h$ with the value h.

The cursor does not "touch" the terrain (P3.2)

The algorithm to determine the position is to use a loop to find the satisfying position, the algorithm is based on several principles:

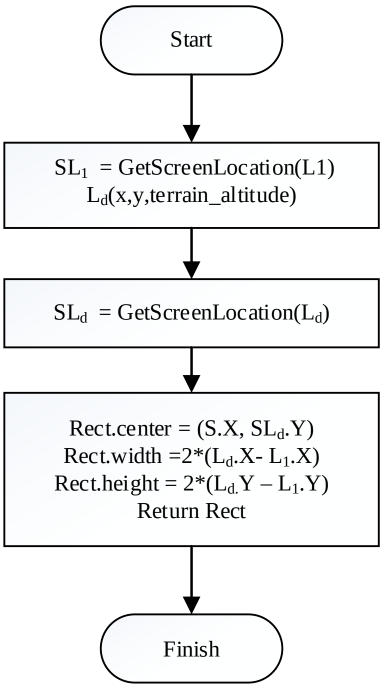
- Traversals the points on the screen, because on the screen there are finite points but in space, there are infinite positions. From the points on the screen that "touch" the terrain will obtain the position in space.

- During the traversal process must check the convergence and update the iteration step to give faster results.

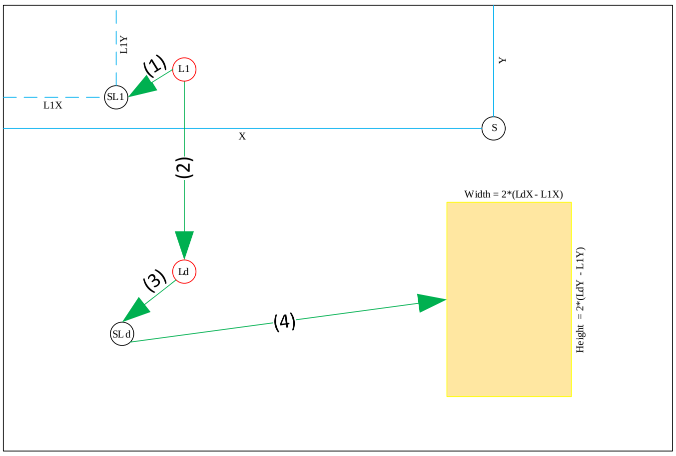
- During the traversal process, the information is saved so that if the results are not found at the end, the best results will be obtained according to a certain criterion.

The algorithm name is "Iterative methods with memory and checking results" follows three big steps:

Step 1: Define the traversal area on the screen. Based on the initial position L_1 (x, y, h), cursor location S(X, Y):



Algorithm diagram



The traversal area is the yellow rectangles on the screen

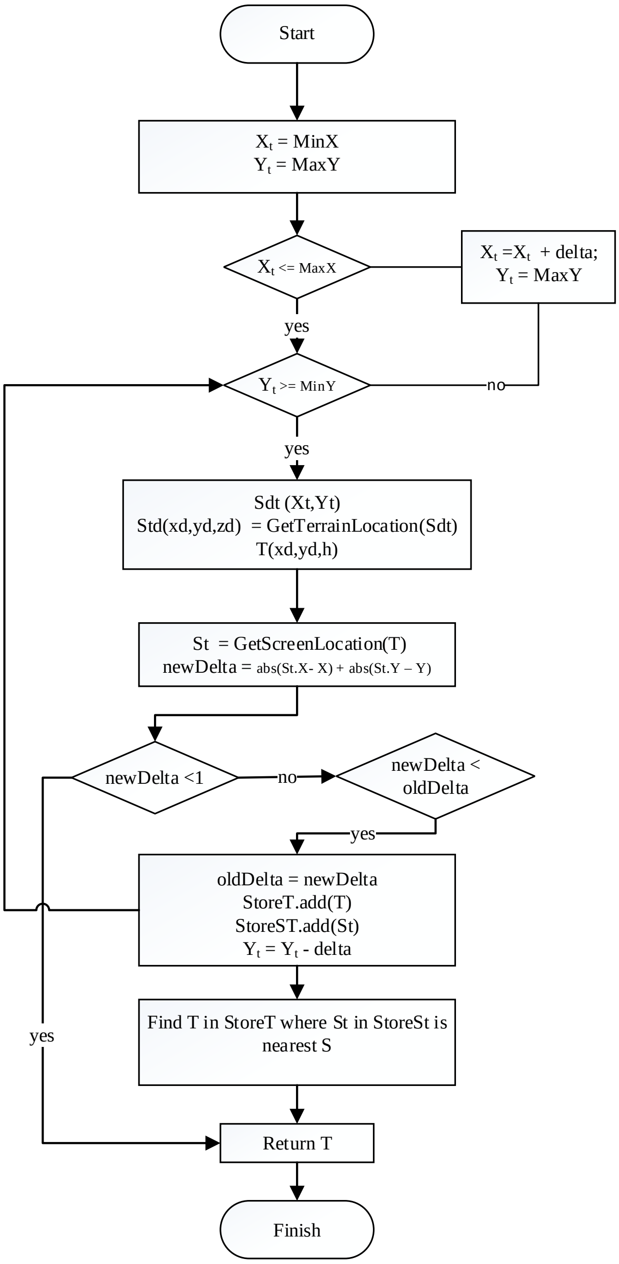
Figure 6. *Describe the algorithm to determine the traversal area.*

Step 2: Perform an iterative traversal area to determine the position.

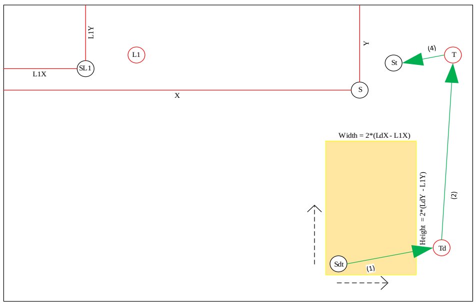
- Nested loop for bidirectional traversing X, and Y in the yellow search area;

- In the loop, must check the convergence condition towards the location S(X, Y). If the convergence is slow or the error is large, adjust the search step, if not, then exit to speed up;

- In the process of repeating the results, so that when the correct result is not found, it must be reviewed to get the best result;



Algorithm diagram



Illustrate a step in a loop

Figure 7. *Illustrate the algorithm to find the position from the search area.*

Step 3: Recalibrate the position.

Usually, the screen area so that the cursor does not "touch" the terrain is the narrow and long area at the top of the application window, for example in the image below is the red rectangular area.

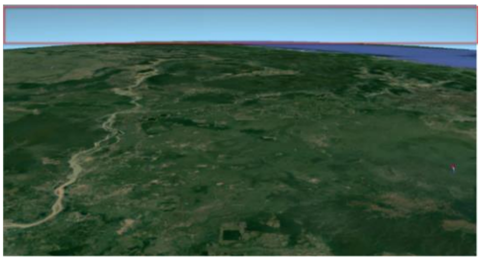


Figure 8. *Illustration of the area where the cursor does not "touch" the terrain.*

In this region the length X is much larger than Y. To operate, users often move in the X direction (to the right or to the left) more. Therefore, in order to ensure that the exact position is found, this step will re-fine the position in the X direction.

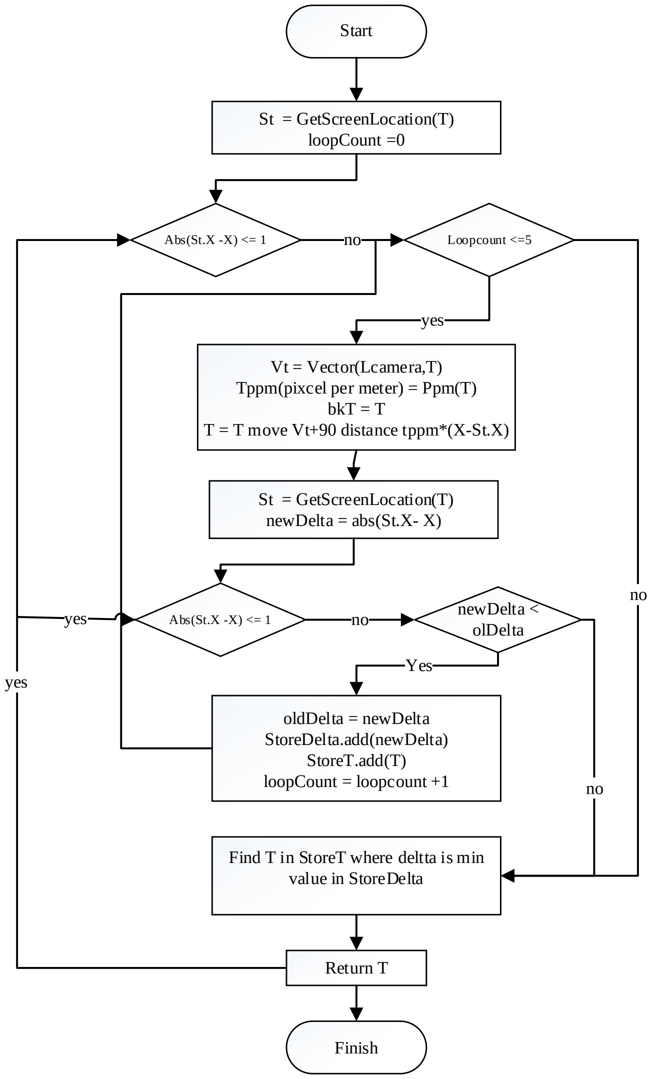


Figure 9. *Algorithm diagram of Recalibrate the position.*

3. APPLY METHOD TO GLOBEDESKTOP SYSTEM

GlobeDesktop is a tool for operations, calculations, and demonstrations of a 3D digital sand table. The functions that use the positioning method include: drawing military symbols, editing symbols, building presentation scripts, measuring, etc. There are many functions that must use the method of determining the position. In this context, the application of the method of job positioning in some typical functions will be presented.

3.1. Drawing military symbols

Select a symbol or a combat object in the military symbology, and move the cursor over the terrain to select one or more suitable locations to place the symbol. Each symbol in the symbol set:

- Group of symbols: group of branches of symbols, type of symbols can be determined: in the air, at sea, on land;

- Initial altitude/draught of object or symbol: Δh;

- Size of object or symbol: Size (w,h,d);

For the group of symbols in the air, on land: Apply according to the case (P1) where the initial altitude is determined: $z = h/2 + \Delta h$, the method of determining the altitude is the altitude above the terrain surface. For groups of signs at sea, depending on the position of the symbol placement:

- If the object is placed on the terrain, the same applies to the case (P1) with the height $z = h/2 + \Delta h$. Note, the method of calculating the altitude, in this case, is the altitude on terrain.

- If the object is placed into the sea with a draught, then apply the case (P1) with the height $z = \Delta h$, the method of calculating the altitude is on the sea. The figure below shows how to determine the height in this case.

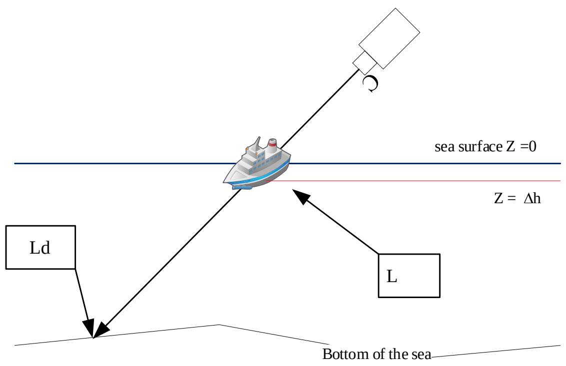


Figure 10. *Illustrate operational position at sea.*

3.2. Editing symbols

Use the cursor to change the position, direction, size, shape, color, etc. of a symbol. Apply the method of determining the position in some cases such as:

- Change altitude: Apply according to the case (P2) the mark position is the current position of the symbol, move the cursor up/down to change the altitude.

- Change position:

• For symbols in the air or at sea: Apply according to the case (P3) move the cursor to determine the new position of the symbol and keep the altitude, the mark position is the initial position of the symbol.

• For symbols on terrain: Apply according to the case (P1) to determine the position.

- Change direction: Use the cursor to move the direct control of the symbol. Apply case (P3) to determine the new position of the direction control, thereby determining the rotation angle.

- Change symbol size: Use the cursor to move the symbol's size determination control.

• For height change, apply case by case (P2).

• For a change of length or width, apply case by case (P3).

3.3. Topography measurement

Use the cursor to move and select one or more positions to calculate: distance, height, area, slope, contour line, etc. The method of determining the position, in this case, is usually case-by-case. (P1) terrain "touching" positions are operational positions. For height measurement function:

- Use the cursor to click on the terrain to get the mark position.

- Move the cursor up/down relative to the mark position, determine the corresponding position of the cursor according to the case (P2);

4. EXPERIMENT AND RESULT

There are two important criteria to evaluate the method of determining location: The average time from the start to the result of the calculation of the position. In operation, the time for user acceptance should be less than 200 milliseconds. The deviation of the result when on the screen compared to the cursor location. In case there is a position satisfying the cursor location, the found result has an acceptable deviation from the pointer of fewer than 10 pixels in each direction X, and Y.

4.1. Experimental environment, scenario

Experimental Environment: The method has been applied in the GlobeDesktop system, so use this software and install additional tools to measure the time and accuracy of the method. The specific environment is described in the table below:

Software	Terrain	Computer info	
		OS	Hardware info
GlobeDesktop	*Quarter: Cam Ranh Airport* *Commune: Cam Hai Dong* *District: Cam Ranh* *Province: Khanh hoa*	*Windows 10 Pro N, version 21H2, 64 bit.*	*Processor:Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz.* *Ram: 16.0 GB*

Experimental scenario: Implement operational scenarios that apply the following cases P1, P2, and P3 specifically:

- Experimental case P1: By moving the ground symbol.

- Experimental case P2: By altimeter function. Split by two locations of the cursor "touching" the terrain and not "touching" the terrain.

- Experimental case P3: By moving the position of the symbol in the sky. Split by two locations of the cursor "touching" the terrain and not "touching" the terrain.

The altitude and pitch of the camera (figure 1) also affect the speed and accuracy of the method. Therefore, the experimental scenario must experiment with the camera heights corresponding to the display level: Quarter: 500 meters, commune/ward: 4,000 meters, district/district: 60,000 meters, province/city: 100,000 meters. And camera pitch: 20-0, 30-0, 50-0, 90-0. With each information, Camera performs 10 measurements and then takes the average of the results.

4.2. Results

Table 1. *Test results in case of P1.*

No	Camera info		Time (millisecond)	Error distance (pixel)	
	Altitude (meter)	Pitch (degree)		Width	Height
1	500	20	~ 0	1.6	0.5
2		30	~ 0	2	0.2
3		50	~ 0	3	0.05
4		90	~ 0	1.4	0
5	4000	20	~ 0	1.6	0.02
6		30	~ 0	1.2	1.2
7		50	~ 0	1.5	1
8		90	~ 0	1.6	5
9	60000	20	~ 0	3.4	2.3
10		30	~ 0	4.1	2.3
11		50	~ 0	5.6	1
12		90	~ 0	7.1	0.03
13	100000	20	~ 0	2	2.2
14		30	~ 0	2.3	3.2
15		50	~ 0	5.7	1.2
16		90	~ 0	6.1	0.9

Table 2. *Test result in case of P2.*

No	Camera info		Time (millisecond)	Error distance (pixel)
	Altitude (meter)	Pitch (degree)		Height
1	500	20	~ 0	6.5
2		30	~ 0	1
3		50	~ 0	2.2
4	4000	20	~ 0	4.1
5		30	~ 0	7.2
6		50	~ 0	7.3
7	60000	20	~ 0	4.3
8		30	~ 0	7.1
9		50	~ 0	6.4
10	100000	20	~ 0	6.3
11		30	~ 0	5.6
12		50	~ 0	7.6

No	Camera info		Time (millisecond)	Error distance (pixel)
	Altitude (meter)	Pitch (degree)		Height
1	4000	20	2.9525	0
2		30	1.994	0
3		50	1.9956	0
4	60000	20	3.975	0
5		30	2.9583	0
6		50	1.9961	0
7	100000	20	7.9119	0
8		30	0.9973	0
9		50	1.995	0

The cursor does not "touches" the terrain

The cursor "touches" the terrain

No	Camera info		Time (millisecond)	Error distance (pixel)	
	Altitude (meter)	Pitch (degree)		Width	Height
1	500	20	~ 0	0	0
2		30	~ 0	7.1	0
3		50	~ 0	8.7	3
4		90	~ 0	3.5	0
5	4000	20	~ 0	7.4	0
6		30	~ 0	7	0
7		50	~ 0	4.5	0
8		90	~ 0	0	0
9	60000	20	~ 0	0	0
10		30	~ 0	0	0
11		50	~ 0	0	0
12		90	~ 0	0	0
13	100000	20	~ 0	0	3.1
14		30	~ 0	9.6	0
15		50	~ 0	0	0
16		90	~ 0	0	0

The cursor "touches" the terrain

Table 3. *Test result in case of P3.*

No	Camera info		Time (millisecond)	Error distance (pixel)	
	Altitude (meter)	Pitch (degree)		Width	Altitude (meter)
1	4000	20	140	0	3
2		30	156	0	4
3		50	195	0	8
4	60000	20	120	0	5
5		30	37	0	2.5
6		50	47	0	9.7
7	100000	20	27	0	14
8		30	5	0	9.7
9		50	14	0	9.2

The cursor does not "touches" the terrain

4.3. Disscusion

Test with normal configuration computer, terrain data from detail to overall. Most of the tests met the criteria for time and accuracy.

All trials met the time criteria. For the case where the cursor hits the ground, the computation time is approximately 0. Because no loops are required, only mathematical functions are performed. For the cases where the cursor does not touch the ground, the calculation time in the case of P3 is higher but still meets the requirements.

The majority of tests met the accuracy criteria. Some results are inaccurate due to the cursor speed being too fast, continuing to apply the method will give more accurate results.

5. CONCLUSIONS

The article has analyzed the practical business to divide it into 3 business cases: P1, P2, and P3. All operations in the 3D digital sand table are referred to in these three cases. Simple operations usually use 1 of these 3 cases. Complex operations are a combination of these three cases.

The article also proposes a method to determine the operational location in these 3 cases. For cases P1, P2.1, and P3.1, it is only necessary to apply common mathematical formulas to be able to determine it, so the accuracy and speed are very high. For the case of P2.2, P3.2 uses a tracked iterative algorithm, with a convergence condition check.

The article also builds an experimental scenario to evaluate the method. Experimental results in a normal working environment. The method is satisfactory in terms of time and accuracy.

The proposed method has been applied in the GlobeDesktop software of the T3BD system, ensuring business requirements as well as basic performance.

In the next phase, the method needs to be studied and improved in the following directions: Business research to re-model use cases in a more scientific and general way. Apply intelligent algorithms to improve algorithm speed for cases P2.2, and P2.3. Smooth operation when switching between cases P2.1 with P2.2, and P3.1 with P3.2.

REFERENCES

[1]. Nguyen Duc Dinh, Hoang Van Toan. "*System Design Documentation of T3BD System*.", (2020).

[2]. "*TerraExplorer Programmer Guide – Basic concepts*", Skyline Sofware Inc, (2017).

[3]. Nguyen Canh Hung, Nguyen Duc Dinh, Le Ngoc Tu, "*Application of the skyline technology platform in the construction of 3d digital sand table data for command and consultation work in sea and island warfare*", Academy of Military Science and Technology, (2019).

[4]. Le Van Diep, Hoang Van Toan. "*Real-time distributed data transmission and application in data synchronization of sea and island battle plans based on digital charts and 3d digital sand tables"*, Academy of Military Science and Technology, (2019).

TÓM TẮT

Phương pháp xác định vị trí tác nghiệp trong sa bàn số 3D

Bài báo này đề xuất một phương pháp xác định vị trí tác nghiệp trong sa bàn số 3D. Bài báo đề xuất các tác vụ cơ bản trong tác nghiệp sa bàn số, và xây dựng các thuật toán để xác định vị trí tác nghiệp trong từng tác vụ. Sau đó, bài trình bày việc áp dụng phương pháp đó vào các chức năng cụ thể của phần mềm sa bàn số GlobeDesktop là sản phẩm của đề tài cấp nhà nước đã được nghiệm thu. Cuối cùng bài báo trình bày nội dung thực nghiệm để đánh phương pháp.

Từ khóa: Sa bàn; Sa bàn số 3D; Skyline Terraexplorer Pro; Bản đồ; Bản đồ số.