A method of drawing double arrow symbols with automation support for 2D digital map applications

Mac Van Vien*

Institute of Information Technology, Academy of Military Science and Technology, 17 Hoang Sam, Cau Giay, Hanoi, Vietnam.
*Corresponding author: vienmvit@gmail.com
Received 10 Oct. 2023; Revised 28 Dec. 2023; Accepted 6 Mar. 2024; Published 22 Apr. 2024.
DOI: https://doi.org/10.54939/1859-1043.j.mst.94.2024.139-148

ABSTRACT

This paper addresses the challenges of translating pivotal military symbols like double arrows onto 2D digital maps, particularly when automation is essential. It proposes a comprehensive method integrating data structure, user workflow, and algorithms to generate tactically aligned, balanced, and visually appealing double arrows for military 2D digital maps. Experimental validation affirms its efficacy in meeting tactical requisites and aesthetic standards. Additionally, it outlines avenues for future research to further enhance and broaden the applicability of this method. In essence, this study presents a robust solution for the automated generation of double arrows on 2D digital military maps, marking a significant step in advancing automated symbolization for military planning.

Keywords: Military maps; Military map symbols; Operations planning; Approximation; Artificial intelligence.

1. INTRODUCTION

1.1. Introducing the double arrow in military maps

Military maps are an appropriate way to visually show military operations and other activities [6]. In that arrow, symbols represent military action and other operations, giving us information about tactical intentions such as the importance of task direction (primary direction, secondary direction), level (force) of tasks (strategic level, campaign level, tactical level, platoon, company, battalion, etc.), deployment position of the mission, mission's goals, the purpose of the mission (attack, counterattack, move the squad, search and hunt, etc.), stages of the mission (which happened or are expected to happen); deployment squad of the mission It can be said that arrow symbols are the soul of the military maps; an arrow in the wrong direction or asymmetrical not only misrepresents the tactical intentions but also affects the overall beauty of the maps. Therefore, drawing arrow symbols is an important issue when building a 2D digital map application. In addition to the general requirements of a 2D digital map application [3], such as satisfying the regulations on mapping consulting work [1], satisfy position accuracy, quick response time, easy to operate, easy to edit. Arrow symbols must also meet the following criteria: True expression of the tactical intentions of the commander, nice shape (proportionate, not over the benchmark).

This paper focuses on the study of generating double-arrow symbols on 2D digital maps with support for automated operations. The double arrow symbol consists of two wings on the right and left of the arrow's axis, representing a tactical intention for attack or counterattack. The wing stroke is darker in the primary direction, and the two wings are farther from the axis than in the secondary direction. The complexity arises from the need to balance the two wings from the deployment position to the mission's target.

Considering technological advancements, the growing interest in using artificial intelligence (AI) to enhance map-making is evident. As noted in [7], AI can significantly automate map generation tasks, offering a simpler, more accurate, and more efficient approach for creating symbols such as the double arrow. This integration of AI not only enhances accuracy and

symmetry but also streamlines the process, reducing the manual effort and time required compared to traditional methods [8, 10]. Our research aims to develop a method that addresses the complexity of double arrow symbols while harnessing the potential of AI for automatic operations in 2D digital map applications [5].

1.2. Concepts and Requirements of the Double Arrow

Concepts

Deployment segment of mission: The start positions of the forces involved in the mission are limited by the endpoints of the two wings of the arrow. The deployment segment width is r_d.

Deployment position of the mission: The center of the deployment segment.

The main axis of the arrow: A line connecting the deployment position to the mission's end goal. The arrow's direction is determined based on this axis, and the total distance between consecutive points on the major axis is denoted as D.

Arrowhead: A pointed arrow with the tip pointing to the mission's end goal. It has a length (d), width (r), and wing length (r_c).

ArrowBody: Two wings on the right and left connect the two vertices of the deployment segment to the inner body of the Arrowhead.

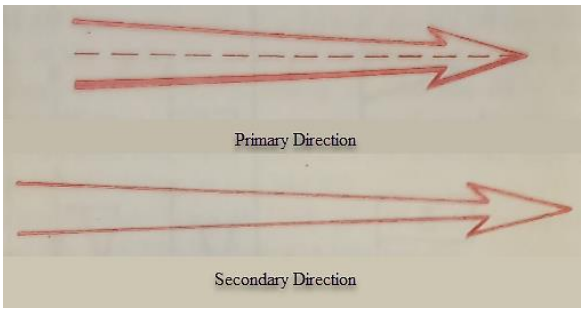

Figure 1. *The double arrow.*

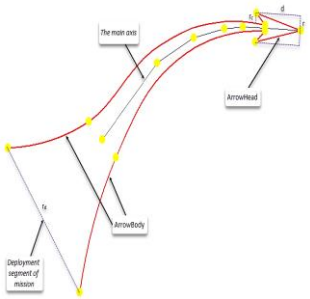

Figure 2. *Concepts in the double arrow.*

Tactical requirements

Tactical requirements dictate that the double arrow symbol must extend from the deployment location to the mission's end goal, reflecting the execution intention of each task. For counterattacks, it points from the deployment position to the target, and for attacks, it points from the deployment position to the captured target.

Technical requirements include ensuring the arrow is in the right direction, balanced, and not exceeding the reference point. Working on 2D digital maps requires simplicity, edit ability, accuracy, and quick response time.

Recommendations for the arrowhead size involve correlation dimensions and guidelines for a balanced shape, considering factors like D/6 to D/9 for r values and adjusting for large D compared to r_d. Correlation dimensions of arrow vertices:

$$r = 1/2*d,\ r_c = 1/3*r \quad (1)$$

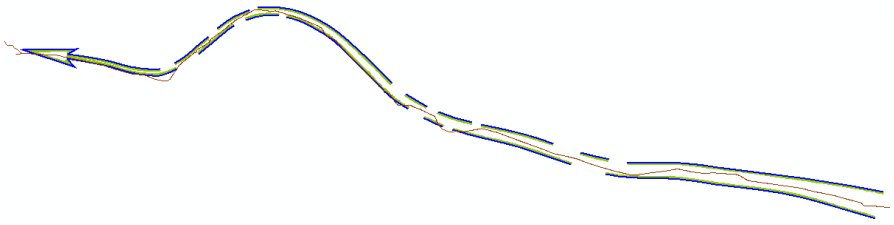

Figure 3. *The example of a double arrow with a very large D.*

1.3. The problem and requirements

Create a 2D map application with an automated double arrow generation algorithm, adhering to tactical, technical, and aesthetic requirements. Evaluate criteria through user feedback for simplicity, shape, and balance. Additionally, experiments will be conducted to ensure the arrowhead points to the mission's end goal, quick response time, ArrowBody symmetry, and adherence to size recommendations.

2. PROPOSED METHOD

2.1. Design of the double arrow on a 2D digital map application

This section presents the structural design (components) of the double arrow symbol and the design of the user workflow for the double arrow. A key requirement of this design is to support automatic operations, which are crucial for drawing double arrows efficiently and accurately. Unlike existing models such as Flow Graphs [4], our algorithm optimally combines symbol manipulation with Cartography/General Staff regulations [1] and double arrow size recommendations (sec. 1.2) to create balanced and aesthetically pleasing double arrows.

2.1.1. The design and user's workflow for the double arrow on a 2D digital map application

The double arrow design consists of three polylines: the left wing, right wing, and Arrowhead. Both wings are symmetrical and curved about the arrow's main axis, while the Arrowhead follows dimensions determined by formula (1).

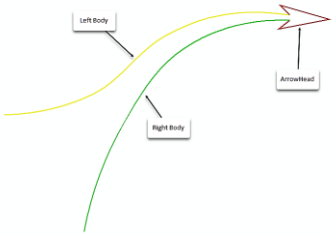

Figure 4. *Design of a double arrow for 2D map application.*

The double arrow operation is divided into two processes: creating a new double arrow and editing an existing double arrow. Therefore, the user's workflow consists of two parts:

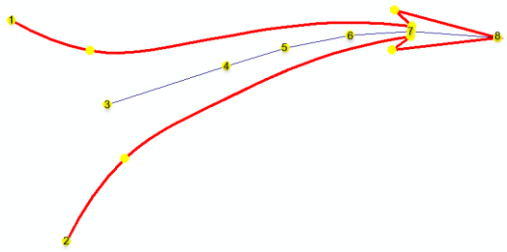

Figure 5. *Drawing a new double arrow.*

- Click on points or enter coordinates (at least 5 points).
- First two points define the arrow's width (Deployment segment).
- Subsequent points form the main axis.
- Last two points create the Arrowhead.

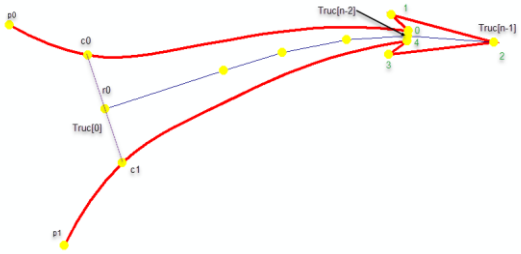

Figure 6. *Editing a double arrow.*

- Retrieve coordinates of left wing, right wing, and Arrowhead for editing.
- Use the mouse to adjust corresponding edit points.
- p0, p1 change deployment segment width.
- c0, c1 adjust arrow body width (r0).
- Points in Truc modify the arrow's main axis.
- Points 1, 3, 0, 4 control Arrowhead width.
- Points 0, 4 adjust Arrowhead body width.

This design allows automatic generation and display of double arrows at two levels:

Level 1: User inputs deployment segment and ArrowHead details;

Level 2: System calculates main axis and ArrowHead based on user-entered deployment path, mission goals, and direction priorities.

2.1.2. The parameters of the double arrow

Operation Points (p[n]): User's operation points, where n ≥ 0.

The parameters at the ArrowHead

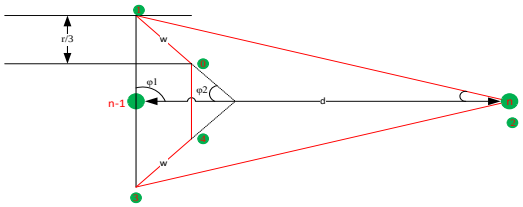

Figure 7. *Parameters of the arrowhead.*

ArrowHead Points (A[5]): A set of 5 points: A0, A1, A2, A3, A4.

Length of ArrowHead (d): determined by the last two operation points: p[n-2], p[n-1].

Width of ArrowHead (r): determined by the width of the two wings or default to d/2.

Large angle of ArrowHead (φ1): determined by the main axis and outer tip of the arrowhead.

Small angle of ArrowHead (φ2): determined by the axis and small wing of the arrowhead.

Length of small wing of ArrowHead (w): determined by distances between outer and inner wings.

The parameters of the ArrowBody

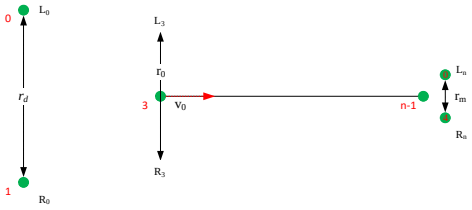

Figure 8. *Parameters of the arrowbody.*

Main axis of the arrow (Truc[]): Operation points excluding the first two: p[2], p[3]…, p[n-1].

Length of the main axis of Arrow (d_T): determined by the total distance in the arrow axis.

Width of Arrow tail (rd): determined by distance between the first two operation points.

Small width of ArrowHead (rm): determined by the distance between vertex A[0] and A[4] of the arrowhead.

Length of the axis at Truc[i] (di): determined by total length from point i to the final point of the main axis.

Direction of the main axis at Truc[i] (vi): determined by direction from point i to point i +1 on the main axis.

Width of ArrowBody at Truc[i] (ri): determined by distance of two symmetrical points on the two arrow bodies through the main axis, based on values of r_d, r_m, d_T, d_i.

2.2. The double arrow operation algorithms

Based on section 2.1.1, the double arrow operation involves two main processes, outlined below:

1. Create a New Double Arrow:

- Use user's operation points to calculate arrow information.
- Generate a new double arrow based on calculated information.

2. *Edit an Existing Double Arrow:*

- Read arrow information of the existing double arrow.
- Utilize the mouse to edit control points for various adjustments: Change arrow tail width; Adjust arrow body width; Modify arrowhead wing width; Adjust arrowhead body width; Shape customization of the arrow.

This article focuses on detailing the process of creating a new arrow, which constitutes the primary content of the double arrow operation algorithm.

2.3. The algorithm for creating a new double arrow

A key feature of our algorithm is its ability to automatically generate double arrows based on the user's workflow. The user simply needs to follow specific steps to define the deployment segment, target point, and arrow direction. Subsequently, the software creates a double arrow adhering to tactical, technical, and aesthetic requirements. This automatic generation not only enhances user efficiency but also ensures uniformity and precision in representing military operations. Here, we outline the algorithm for the automatic generation of double arrows:

Input: Set of n user input points p[n].

Output: Draw a double arrow comprising the left wing, right wing, ArrowHead, and the arrow body, meeting specified requirements.

2.3.1. Check Input

The algorithm checks whether the user input points p[n] can create a double arrow or not. To create a double arrow, the number of points n ≥ 5, the edges created by consecutive points do not intersect each other and do not cut into the arrow tail.

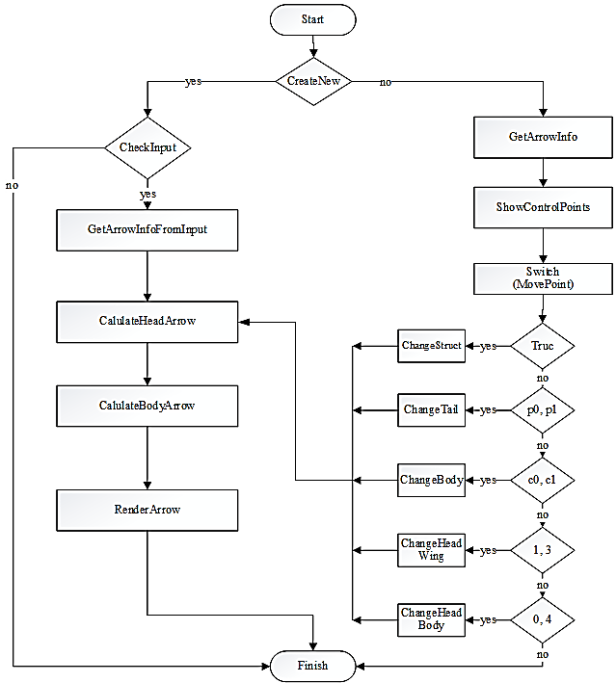

Figure 9. *The algorithms of double arrow operations.*

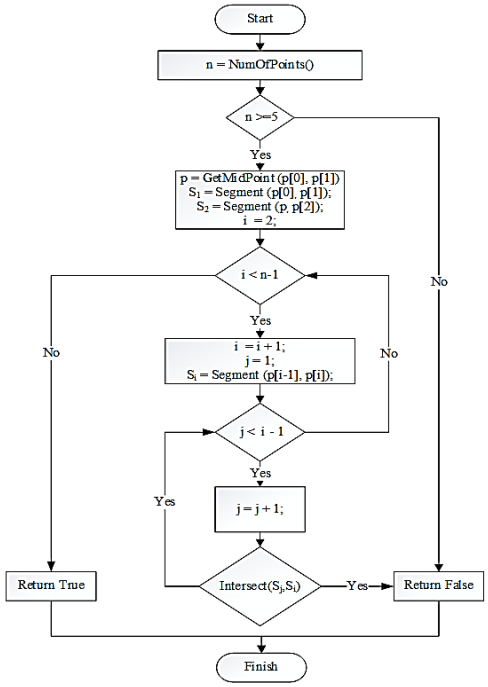

Figure 10. *The Algorithm for checking input.*

2.3.2. Get double arrow information from input

From the input points, the user must calculate the double arrow information to generate the components of the arrow. The algorithm calculates the general information of the arrow, including the beginning points of the left wing, the right wing, and the main axis of the arrow. The algorithm to get general information of the double arrow from points in p[n] is described as shown below.

2.3.3. Calculate arrowhead

From the set of points on the main axis (Truc) and the parameters of the double arrow (if any), perform calculations to create the points of the arrowhead. The set of arrowhead points is stored in the array A [5]. The algorithm diagram for calculating the points of the arrowhead is described as follows:

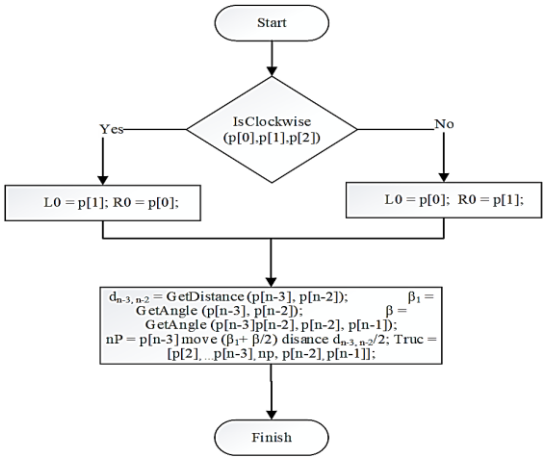

Figure 11. *The algorithm for getting arrow information.*

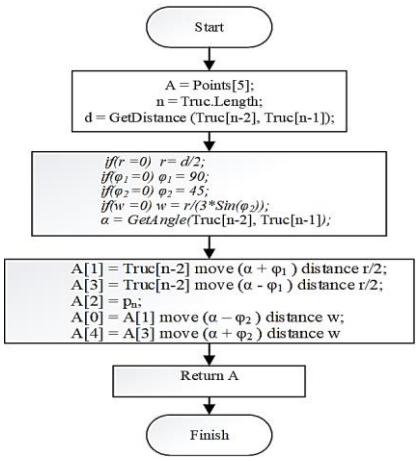

Figure 12. *The algorithm for calculating the arrowhead.*

2.3.4. Calculate arrow body

Calculate the points of the left wing (L[]) and right wing (R[]) of the arrow body, based on the set of points on the main axis (Truc), the five points forming the arrowhead (A), and the specified double arrow parameters. When creating a new double arrow, the crucial step is to calculate the arrow body width (r0). The method involves measuring manually crafted battle plans' double arrows and using regression analysis to determine r0. Regression equations are derived from existing battle plans with given variables: $x = d_T/r_d$ and $y = r_d/r_0$.

Table 1. *Measurement results of double arrows in existing battle plans.*

x	~ 2.0	~ 3.0	~ 5.0	~ 7.0	~ 9.0	~ 20.0
y	*~ 0.35*	*~ 0.4*	*~ 0.5*	*~ 0.6*	*~ 0.65*	*~ 0.7*
Number	*5*	*5*	*5*	*5*	*5*	*5*

Using some regression calculation methods [9], we obtain the regression equations that best fit the data, with the results shown in the table below:

Table 2. *The results of Function approximation with regression analysis.*

No	Regression Type	Equation	Correlation (%)
1	Linear regression	$y = 0.0182x + 0.3939$	85.12
2	Quadratic regression	$y = -0.0022x^2 + 0.0685x + 0.2183$	99.90
3	Cubic regression	$y = 0.0000x^3 - 0.0023x^2 + 0.0689x + 0.2177$	99.90
4	Power regression	$y = 0.2915x^{0.322}$	93.99
5	ab-Exponential regression	$y = 0.3967 \cdot 1.0352^x$	78.82
6	Logarithmic regression	$y = 0.2407 + 0.1666 \cdot \ln x$	97.23
7	Hyperbolic regression	$y = 0.7138 - 0.8095 / x$	96.00
8	Exponential regression	$y = e^{-0.9245 + 0.03466x}$	87.82

The value of r_0 is determined by the quadratic regression equation:

$$r_0 = \frac{r_d}{-0.0022x^2 + 0.0685x + 0.2183}; \qquad x = \frac{d_T}{r_d} \tag{2}$$

Subsequent r_i values are calculated by:

$$r_i = r_m + \frac{d_i}{d_T}(r_0 - r_m); \qquad i \in [1; Truc.Length] \tag{3}$$

2.3.5. Render double arrow

Render the Double Arrow by determining color, drawing style, line thickness, and fill color based on the task's characteristics, stages, and goals. Geometric structure rendering is performed for the three objects: left wing, right wing, and arrowhead. The technical requirements dictate the shapes of these objects:

The arrowhead is represented using straight lines. The left and right wings of the arrow are "cubic spline interpolations" with 150 interpolation points, a value experimentally chosen to balance calculation speed and aesthetic appeal.

GisDesktop, a military 2D map operation software [2], successfully applies the double arrow drawing method. Positive user feedback emphasizes the aesthetic, symmetrical appearance of arrows, and the software's user-friendliness.

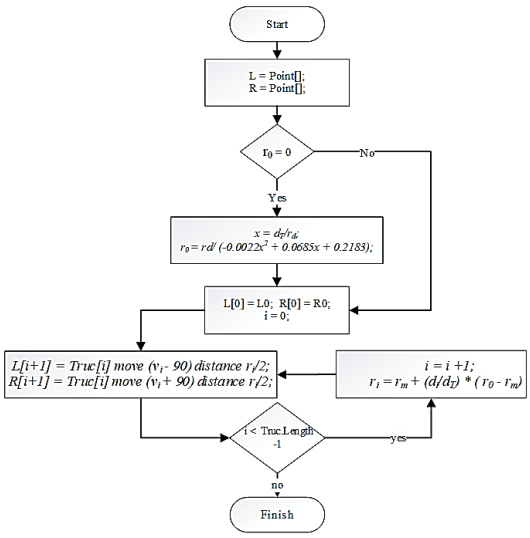

Figure 13. *The algorithm diagram for generating the double arrow body.*

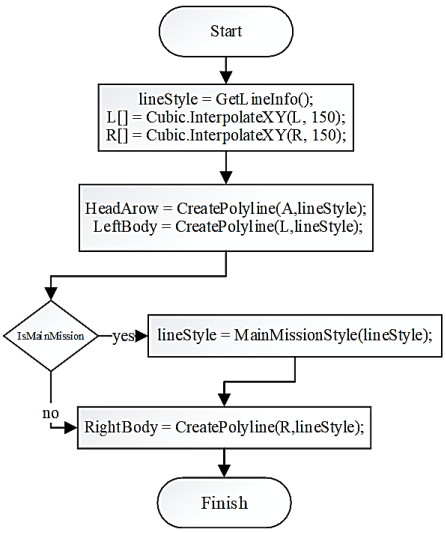

Figure 14. *The algorithm diagram for rendering double arrows.*

3. APPLICATION AND EXPERIMENTS

3.1. Experimental criteria

Criteria evaluation involves measuring various aspects using GisDesktop, as outlined below:

Table 3. *Measurement method and the criterion's pass condition .*

No	Sign	Criteria name	Pricing method and eligibility criteria	Unit
1	Criteria 1	The arrowhead points to the mission's end goal	Measure the distance from the arrowhead top to the final position of the main axis: < 5*linewidth	Pixcels
2	Criteria 2	Quick response time	Count the time from the end of the drawing operation to the completion of the arrow display: <= 500	Milliseconds
3	Criteria 3	The arrowbody is symmetrical about the main axis	|DistancOf (c0 Truc [0]) – DistancOf (Truc[0] c1)| < 5* linewidth (figure 5).	Pixcels
4	Criteria 4	The size of the arrowhead follows recommendation (c.1)	|d – 2*r| < 3* linewidth (figure 1)	Pixcels

3.2. Experimental environment, scenario

Experimental environment: The double arrow drawing method has been applied in the GisDesktop system, so we use this software and install a stopwatch and a ruler to measure the time and accuracy of the method. The specific environment is described below:

Table 4. *Experimental environment.*

Software	**Area**	**Computer info**	
		OS	**Hardware information**
GisDesktop	East Sea	Windows 11 Pro N, version 22H2, 64 bit. Display resolution 1920 x 1080	Processor:Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz. Ram: 16.0 GB. Graphics card: support full HD.

Experimental scenario: Draw symbols at a map scale with line width from 1 to 7 pixels, the number of points on the main axis ranging from 5 (the minimum) to 15. Perform counting of drawing time, measuring the size and shape of the arrows at different map display scales.

Scenario 1: Draw 70 symbols at map scale 1000000; Measure distances at map scales 1000000 and 100000.

Scenario 2: Draw 70 symbols at map scale 100000; Measure distances at map scales 100000 and 25000.

3.3. Results

Table 5. *Test results of experimental 1.*

Line width	**Criteria 1**		**Criteria 2**					**Criteria 3**		**Criteria 4**	
	1 000 000	**100 000**	**Number of input points**					**1 000 000**	**100 000**	**1 000 000**	**100 000**
			5	**7**	**9**	**11**	**15**				
1	< 5		220	180	256	350	357	< 5		< 3	
	1.2	4.5						0.4	2	0.2	2.5
2	< 10		125	280	255	387	420	< 10		< 6	
	2.2	5.1						0.5	3.7	0.1	2
3	< 15		131	234	321	411	359	< 15		< 9	
	3.7	7.6						0.2	4	0.6	3
4	< 20		201	198	254	312	412	< 20		< 12	
	8	9.5						1.6	7.9	1.2	5.1
5	< 25		180	189	233	299	401	< 25		< 15	
	7.5	9.9						1.8	7.2	1.3	6.5
6	< 30		137	256	270	306	327	< 30		< 18	
	9.1	12.5						1.5	6.1	2.6	6.9
7	< 35		177	233	242	347	476	< 35		< 21	
	9.5	15.2						1.2	7.3	3.5	8.2

Table 6. *Test results of experimental 2.*

Line width	**Criteria 1**		**Criteria 2**					**Criteria 3**		**Criteria 4**	
	100 000	**25 000**	**Number of input points**					**100 000**	**25 000**	**100 000**	**25 000**
			5	**7**	**9**	**11**	**15**				
1	< 5		134	142	307	349	380	< 5		< 3	
	1.3	2.6						0	0.4	0.4	2.4
2	< 10		77	233	337	405	487	< 10		< 6	
	1.4	3.2						0.1	1.7	0.1	1.4

Line width	Criteria 1		Criteria 2					Criteria 3		Criteria 4	
	100 000	25 000	Number of input points					100 000	25 000	100 000	25 000
			5	7	9	11	15				
3	< 15		10	23	21	37	33	< 15		< 9	
	2.1	3.7	8	7	3	6	1	0	0.2	1.2	3.8
4	< 20		12	25	38	44	44	< 20		< 12	
	2.8	4.7	4	4	9	3	8	2	4.7	0.5	2.5
5	< 25		13	28	31	29	47	< 25		< 15	
	2.5	5.3	0	6	7	9	8	0.2	5.5	0.8	3.7
6	< 30		15	22	28	34	30	< 30		< 18	
	5.7	5.9	4	7	8	2	1	0.9	3.5	1.9	7.3
7	< 35		18	26	25	42	38	< 35		< 21	
	4.1	7.4	1	4	1	8	9	0.6	2.4	3.1	7.7

3.4. Discussion

The software achieves precise results through high-resolution measurements. Criteria 1 exhibits minimal errors, usually less than one-third of the threshold. Criterion 2's results align with the verified interpolation points. For criteria 3 and 4, negligible errors arise only when zooming in at different scales, well within acceptable limits.

The experiments showcase the method's adaptability in diverse conditions, affirming its practical utility beyond theoretical contexts. Its consistent adherence to criteria highlights real-world usability regarding accuracy, response time, symmetry, and size. The method's success suggests future potential, such as integration with advanced technologies like voice recognition, opening avenues for further development. Its proven efficacy in military and geographic applications underscores its pragmatic functionality, solidifying its value in real-world scenarios. The experiments conclusively demonstrate the software's reliability in meeting specified criteria, with negligible errors well within acceptable ranges, emphasizing its robustness and practical utility under various conditions.

4. CONCLUSIONS

This paper has set out to develop a method dedicated to the creation and editing of double arrow symbols on 2D digital maps, specifically designed for tactical applications. An analysis of the broader significance of tactical arrow symbols, with a particular emphasis on double arrows, has been presented. The document introduced key concepts and criteria related to tactics, techniques, and aesthetics essential for constructing an effective arrow application. A comprehensive method for operating double arrow symbols on 2D digital maps, encompassing structural design, operational processes, and creation algorithms, was proposed.

The implementation of this method in the GisDesktop software yielded successful results, as validated by experimental outcomes that unequivocally met the predefined criteria. A notable feature of this method is its proficiency in automatic operations, a crucial aspect of ensuring the swift and accurate depiction of double arrow symbols. This not only enhances the user experience but also guarantees consistency and precision in representing military operations on digital maps.

Looking ahead, future developments should concentrate on refining algorithms for editing double arrow symbols and exploring the integration of advanced artificial intelligence techniques, such as voice recognition [8], or leveraging YOLO-G for enhanced military target detection [10]. These enhancements aim to elevate the effectiveness and versatility of the method, further establishing its value in creating and manipulating double arrow symbols on 2D digital maps.

In summary, the proposed method has conclusively demonstrated its effectiveness and

feasibility in creating and editing double arrow symbols on 2D digital maps. The integration of future enhancements, particularly the infusion of artificial intelligence, holds promising potential to enhance and refine this method, solidifying its standing as a sophisticated and indispensable tool within the realm of tactical mapping applications.

REFERENCES

[1]. Department of Cartography/General Staff, "*Map consulting work*", (2019).

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

[3]. D Swann, "*Geographical Information Systems*", Second Edition, Abridged, (2005).

[4]. G. Wallne, "*Enhancing Battle Maps through Flow Graphs*," *IEEE Conference on Games (CoG)*, London, UK, pp. 1-4, (2019), doi: 10.1109/CIG.2019.8848052.

[5]. Ferenc FAZEKAS, "*Application of Artificial Intelligence in Military Operations Planning*", AARMS (21) 2, (2022).

[6]. Kent, A. J., Davies, J., & Davis, M. "*Soviet mapping: Then and now*". The Cartographic Journal, 59(4), 259-263, (2022). https://doi.org/10.1080/00087041.2022.2267282.

[7]. Meerveld, H.W., Lindelauf, R.H.A., Postma, E.O. et al. *"The irresponsibility of not using AI in the military"*. Ethics Inf Technol 25, 14, (2023). https://doi.org/10.1007/s10676-023-09683-0.

[8]. Dang Duc Thinh, Nguyen Chi Thanh, et al. "*A voice search engine for military symbols to enhance the drafting of operational plan documents on digital map*". Journal of Military Science and Technology 87:40-49, (2023). DOI:10.54939/1859-1043.j.mst.87.2023.40-49.

[9]. Minshuo Chen, Haoming Jiang, Wenjing Liao, Tuo Zhao, *"Nonparametric regression on low-dimensional manifolds using deep ReLU networks: function approximation and statistical recovery",* Information and Inference: A Journal of the IMA, Volume 11, Issue 4, pp. 1203–1253, (2022), https://doi.org/10.1093/imaiai/iaac001.

[10]. L. Kong, J. Wang and P. Zhao, "*YOLO-G: A Lightweight Network Model for Improving the Performance of Military Targets Detection*," in IEEE Access, vol. 10, pp. 55546-55564, (2022), doi: 10.1109/ACCESS.2022.3177628.

TÓM TẮT

Phương pháp vẽ mũi tên kép có hỗ trợ tự động hóa cho ứng dụng bản đồ số 2D

Bài báo này nghiên cứu vấn đề tự động hóa việc vẽ các ký hiệu quân sự trên bản đồ số 2D, cụ thể là ký hiệu mũi tên kép để biểu diễn các hành động quân sự. Bài báo đề xuất một phương pháp toàn diện, bao gồm cấu trúc dữ liệu, quy trình làm việc của người dùng, và các thuật toán sinh các thành phần của ký hiệu mũi tên kép. Phương pháp này có thể tạo ra các ký hiệu mũi tên kép đúng ý định chiến thuật, cân đối và đẹp mắt. Bài báo cũng chỉ ra một số hướng nghiên cứu tiếp theo để cải thiện và mở rộng phương pháp này. Bài báo này đóng góp một giải pháp hiệu quả cho việc tự động hóa việc vẽ các ký hiệu quân sự trên bản đồ số 2D, đánh dấu một bước tiến quan trọng trong lĩnh vực kế hoạch quân sự.

Keywords: Bản đồ tác chiến; Ký hiệu quân sự; Kế hoạch tác chiến; Xấp xỉ hàm; Trí tuệ nhân tạo.