Raster to Vector Map Convertion by Irregular Grid of Heights

Serge Popov, Vadim Glazunov, Mikhail Chuvatov, Alexander Purii
Higher School Of Applied Mathematics and Computational Physics
Peter the Great St.Petersburg Polytechnic University
St. Petersburg, Russia
{popovserge, glazunov_vv}@spbstu.ru, misha@iktp.spbstu.ru, purii.alexander@gmail.com

***Abstract*—The accuracy of the naval map representation determines the quality of route construction when solving the problem of automatic control of ships movement in a difficult navigation environment. Using a high-precision raster map requires a large amount of memory to store data. Vector representation of the map is an alternative approach to the problem of storing data about terrain, it allows to store only the data containing the terrain contours. In this paper we implement and research the algorithms which allow to vectorize the raster maps by the set of isoline slices using irregular grid of heights. These algorithms approximate the map with a set of elevation slices, recursively search for nested isolines on the slice, build and optimize contours of the isolines on the slice. We conduct a study that shows a significant reduction in the amount of stored data in vector form compared to raster format. The proposed format for storing map data will be used to represent realistic landscapes in the tasks of building ship routes.**

I. Introduction

Accurate representation of the naval map is crucial when solving the problem of ships group automatic movement control. To build an accurate ship trajectory, an accurate representation of the depth map should be used [1]. The maps cover huge territories measured in hundreds of square kilometers or more. This requires a large amount of memory to store map data, especially in raster form [2]. Vector format data card is an alternative to raster, it allows data to be stored on the contours of the landscape - isolines of heights (isohips) and depths (isobaths) – on a pre-selected altitude.

By selecting and optimizing contour points located at the same height, the data volume is reduced [3]. This is a significant advantage of the vector representation of the map. An additional advantage is the ability to visualize map fragments when zooming without losing accuracy and display quality.

This approach is widely used when preparing electronic sea [4], [5] and land maps for use in routing systems [6] . The paper offers a technique that allows to form an irregular grid of heights and ensure the selected accuracy of the representation of the vectorized map for different classes of ships. The source data for creating vector maps are raster images: satellite images, aerial photographs, point clouds from lidar altimeters or multipath sonar depth gauges with regular or irregular pixel location [7]. The task of building vector maps is to highlight the contours of the isolines. To solve this problem, segmentation methods are used with subsequent contour construction [8], [9], [10], [11].

The landscape isolines selection at a specific height is the main problem in the vector maps construction. This problem can be formulated as the finding the regions contours in a two-dimensional image.

In this paper we propose the new algorithm for vectorizing three-dimensional raster map with regular grid for the case of an irregular height steps.

II. Related Works

Research on vectorization of raster images are maintained for a long time. However, in most cases, they come down to the image segmentation problem [8], [9], [10], [11] or approximating a raster images by a set of curves [12], [13]. The paper [8] describes the algorithm for obtaining vectorized roof contours of buildings from digital surface model (DSM). The author makes uses the horizontal projection building model with external and internal right angles. For this case, buildings projections are segmented by the region growth algorithm. Vectorized paths are retrieved by looping around the resulting segmented polygons. For each point of the contour, its eight-neighborhood region is considered and compared with reference samples of the distribution of points in the eight-neighborhood connected region. If the distribution in the region coincides with the standard, then it is considered that this point belongs to the contour, otherwise it is isolated. The disadvantage of this approach is that the standards proposed by the author do not allow the contours construction with angles other than straight.

Another approach when vectorizing raster maps is to use pixel vector fields [14]. At the same time, each pixel is associated with a multifactorial spatial and qualitative measure. For example, a quality measure can be an intensity vector of red, green, or blue or another set of criteria if the pixels in question are not painted over with a solid color, but with some texture. In this case, the vector field is described as a set of interaction forces between pixels. The interaction force is described as a vector between two pixels multiplied by a function of the spatial measures of both pixels, multiplied by a function of the qualitative measures of both pixels. It is demonstrated in the work that when using the Euclidean distance as a spatial measure and a

ISSN 2305-7254

three-color vector to determine a qualitative measure the centers of regions of the same color and the boundaries of regions of different colors have about zero vector length which means that these points are reference points of the contour during vectorization.

Another approach to solving the problem of presenting a landscape map can be polygonization of the map, which is a subspecies of vectorization. The paper [9] presents a method using rectangular regulation of areas classified as buildings on a two-dimensional raster map. This approach allows get the result in the form of areas on the map with irregular mesh.

The article [15] describes the Potrace algorithm, which vectorizes a black and white raster image by constructing closed loops on the original image. When building a path, Potrace selects the next point based on 2x2 pixel reference neighborhoods around the starting point. When a circuit is closed, its color is inverted, followed by the recursive algorithm which allows find nested contours. The Potrace algorithm is similar to the approach proposed by the authors; the differences are in the contouring algorithm.

An algorithm for constructing vectorized images based on line-by-line scanning of a large-sized raster image in one pass is given in [16]. The article discusses the method of constructing vectorized images by the bidirectional chain boundary tracing algorithm. The bidirectionality algorithm considers only the left and bottom pixels when constructing the contour. The author introduces two types of connections between pixels: virtual and real. If the previous pixel matches the current, then the connection is considered virtual, otherwise it is considered real. Accordingly, all pixels except the boundary ones have 2 more incoming connections from above and to the right. When sequentially processing pixels, pixels that contain at least one real connection are referred to the contour. Image is vectorized using polygon construction. Polygon contouring starts from the upper left corner and ends in the lower right. If the point contains at least one real connection, then it is part of the circuit; otherwise, it is an internal point. This algorithm allows limiting the simultaneous consideration by only two lines from the original image: current and previous. It uses small RAM buffers and takes into account the specifics implementations of cameras for the visual range view of artificial Earth satellites. This solution disadvantages is the hierarchy lack of the constructed contours.

An alternative way to represent the source data in the constructing contours problem can be vector representations for example in the form of a Delaunay triangulation network [17]. The source data is presented in the point cloud form which executes the triangulation operation and the Delaunay network is formed. The considered equivalent edges set is obtained by sectioning the triangulation network surface with the plane at a given height. For each edge there is the intersection point with the plane and closed and open contours are built on the points resulting set. From the obtained points the two-dimensional Delaunay triangulation is constructed. The resulting edges list is recursively scanned. If the contour closes on itself or on the border, then the contour is considered constructed. The operation continues until edges remain in the original set.

III. Main Part

A. Task

The main goal of the paper is to develop algorithms of raster map vectorization by height slices with isolines for irregular grid of heights.

Input:

1) raster map of heights;
2) irregular grid of heights.

Output:

vectorized map.

B. Vectorization algorithms

1) Algorithm of surface approximation by set of vectorized height layers with irregular grid of heights: Since result of vectorization original surface is presented as a set of vectorized surface slices for specific set of heights, the problem of surface data preprocessing arises. The aim of preprocessing is to build individual surface slice for each height form irregular grid. Slice is consist of binary values because should only contain information that point at specific height is either surface or water. Thus all points of original surface which height are greater or equal than slice height should be interpreted as surface, otherwise it should be interpreted as water. After creating a slice, the contour finding algorithm is invoked. Algorithm of surface approximation shown on Fig. 1.

Algorithm can be efficiently parallelized on data level because contour finding algorithm is using only slices that have been built individually. Thus contours at each height can be built in parallel.

2) Contour finding algorithm: Algorithm iteratively bypasses the slice matrix until it finds a first point which type is surface. After that recursive contour building function is invoked with surface as activation value. Recursive contour building function invokes the contour building algorithm. After finding a contour the process of inner contours finding starts. Wave algorithm finds all points of all inner contours inside the borders of current contour. The point is treated as inner contour point if it lays inside current contour, have different type than current contour. All inner contours points are sorted by X then by Y by ascending to satisfy contour building function requirements. For each point of inner contours recursive contour building function invokes if this point is not contained in already built inner contour. For inner contours activation value inverts. After all inner contours have been built or if there are none, points of current contour and its inner part change their types to opposite. Wave algorithm is used for change points type. After recursive contour building function returns the result it is added to list of isolines on current height. Iterative bypasses proceeded until next point with surface value. The algorithm finish when all matrix has been scanned. Contour finding algorithm shown on Fig. 2.

Proposed recursive contours building approach makes possible to find contours of any nesting depth and to represent isolines as hierarchy of nested contours. Isolines types

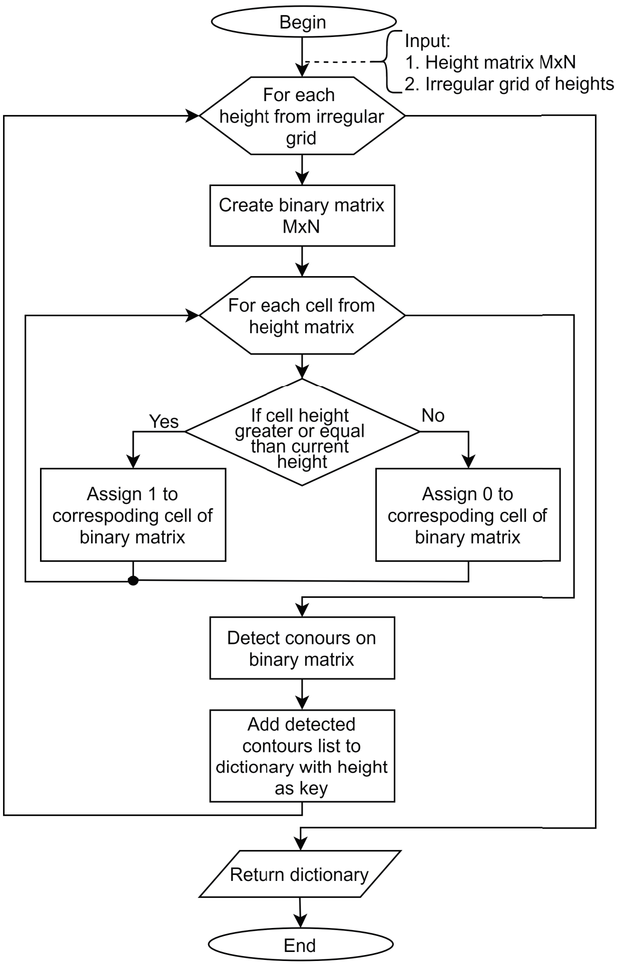

Fig. 1. Block-schema of surface approximation algorithm

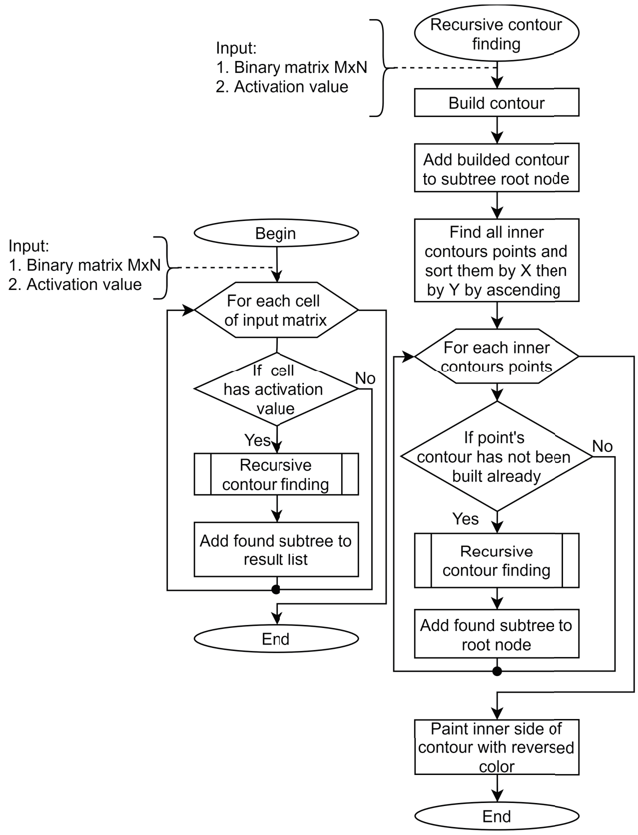

Fig. 2. Block-schema of contour finding algorithm

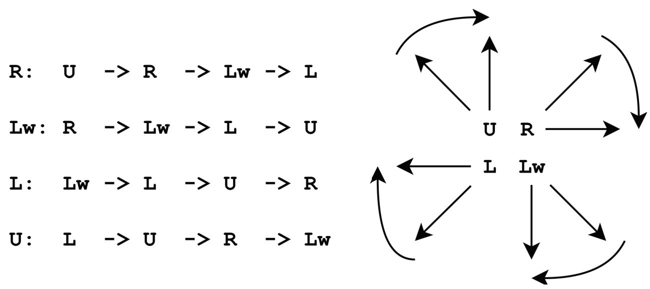

Fig. 3. Pick next point rules and their traversal orders

are cyclically changing surface and water based on a depth of nesting. Similar approach has been used in paper [15] for vectorization of raster images. Contour finding algorithm returns subtree that has a current contour and isoline type in the root node and an inner isolines as branches. Thus that kind of structure can be called as n-ary tree where each layer of nested depth defines a type of stored contour.

3) Contour building algorithm: Algorithm bypasses whole contour for linear time. It assumes that first point of contour is its top left point. Contour is bypassed clockwise and pick of next point of contour is made accordingly to the proposed rules. All of the built contours are closed.

There are 4 base rules or directions that are considered during the contour building: right (R), lower (Lw), left (L), upper (U). These four rules as it is shown on Fig. 3 cover 8-neiborhood.

However, usage of these base rules forbids building any contours but convex what are not cover all possible contour shapes.

In the purpose of solving the problem four additional shuffle rules are proposed. Each of this rules shuffle base rules traversal orders based on previously selected base rule to provide the ability bypass a contour with any possible shape. If previously selected rule is R then traversal order for the next step must be U, R, Lw, L. If previously selected rule is Lw then traversal order for the next step must be R, Lw, L, U. If previously selected rule is L then traversal order for the next step must be Lw, L, U, R. If previously selected rule is U then traversal order for the next step must be L, U, R, Lw.

Base rule implementation order matters. For example, if on previous point selection right or top right point was chosen then during current selection will be attempt to go upper then to go right then to go lower and only then to go left. Contour build finished when next point is equal to start point of contour. Contour building algorithm shown on Fig. 4.

Example of rules implementation presented on Fig. 5.

Proposed approach makes possible to build closed con-

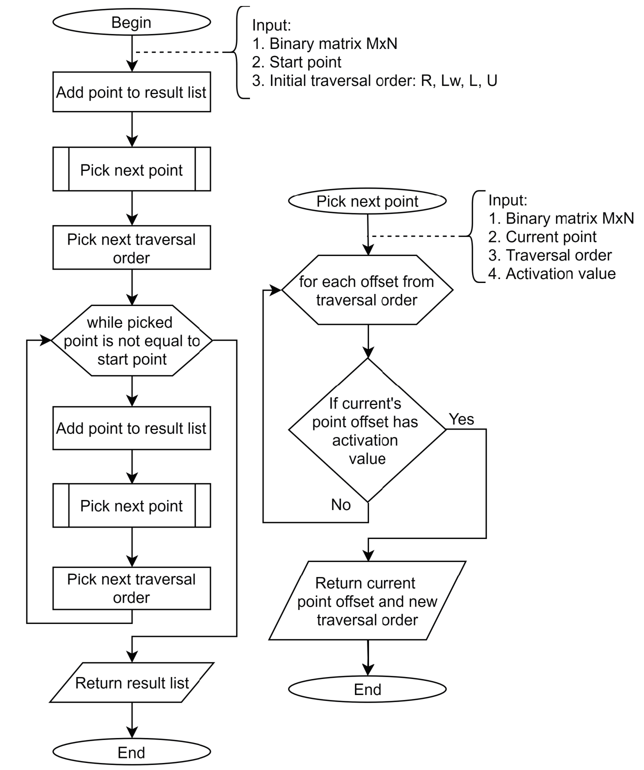

Fig. 4. Block-schema of contour building algorithm

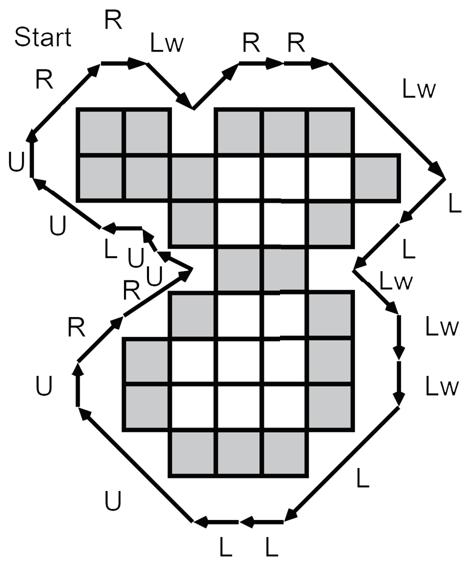

Fig. 5. Contour building example

tours of any shape which is important because shape if isolines on every slice could be arbitrary. Contour building algorithm is used for building either surface or water isolines. However, there are problem with a water contour building. This problem appears when surface contour divides inner and outer water with a 45-degree contour line with 1-pixel width. In that case inner contour water will become part of outer water contour via 8-neiborhood. In the purpose of handling that malfunction during the building of water contours, all points that are separated by diagonal surface contour must be prohibited for consideration.

4) Lossless contour optimization algorithm: Contour that have been built has redundant data like points laying on a straight line. These can be removed without any damage to contour shape. During the contour optimization all points that are laying on the same line should be replaced with start and end points. To ensure that there are no losses only lines that corresponds 8-neiborhood movement directions are considered.

In operability check purposes and to conduct researches of algorithms an application was written in C#. It is capable to generate vector map by vectorising a raster map which is generated using Perlin Noise method. Besides that, program is capable to draw vectorized map. The input to the algorithm is a bitmap in 256 grayscale. The gray level determines the height or depth. At the output of the algorithm, an array of vectorized contours is stored. It's saved in the array of structures containing pairs: height and an ordered array pairs of points. Each point is the boundary of the isoline. Example of graphical output shown on Fig. 6.

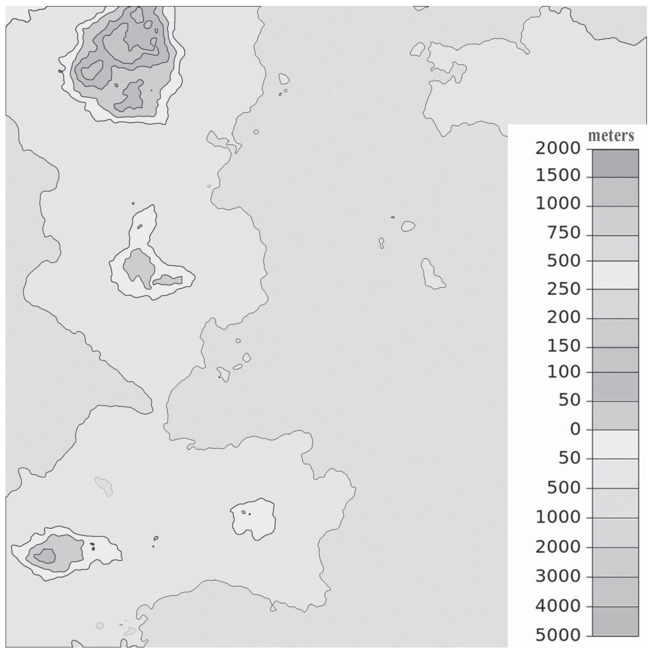

Fig. 6. Example of program graphical output

IV. Research

We have implemented a set of algorithms that approximates a three-dimensional raster map with a set of slices of height isolines. The main reason for using this method is to reduce the amount of stored data, so it is important to investigate the relationship between the amount of data, the time it takes to build isolines, and the distribution of the height approximation error on the number of slices and the size of the raster map. Using the software implementation of the Perlin Noise method, we generated the landscapes of the original raster maps. The proposed vectorization algorithms are designed to be used to create marine vector maps, so we chose the height of the conditional sea level to construct the slices. At this altitude, the ratio of land surface area to water is approximately 1 to 10, which brings the content of the generated maps as close as possible to real sea maps for special navigation conditions. To approximate at a given sea level, we used an irregular grid of heights with variable increments from 0.4 to 4 meters in the range of heights from -40 to 40 meters. The selected range provides accurate source data for computation the trajectories of ships with a large range of dimensions.

A. Parameters and constants of the experiments

Table I. PARAMETERS AND CONSTANTS OF THE EXPERIMENTS

Parameters of the experiments	
Map size, m×m, px.	100×100, 200×200, 400×400, 600×600, 1000×1000
Number of slices, n, pcs.	10, 20, 40, 80, 100
Number of iterations, times	20
Hardware characteristics	
CPU	AMD Ryzen, 3.6GHz, 8 cores/16 threads
RAM	16GB

B. Experiment 1

1) Experiment statement: Determine the correlation between the volume of stored raster and vector data, the size of the original raster grid, and the number of height slices.

2) Experimental results: The amount of data to store in raster form increases quadratically from the linear size of the map, but does not depend on the number of isolines slices. The amount of data to store in vector form increases linearly, depending on both the map size and terrain complexity, and the number of slices. As a result, the amount of stored data differs several times on large maps. A graphical representation of the experiment results is shown in Fig. 7.

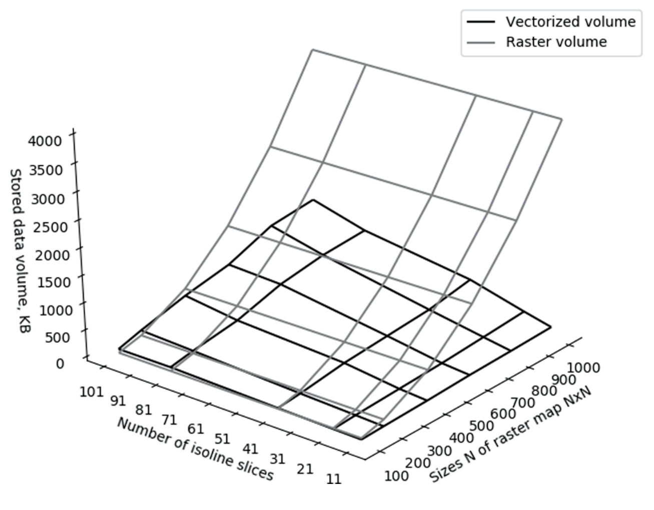

Figure 7. Graph shows the dependence of the amount of stored data on the number of slices and the size of the raster map

The results of the experiment depend on the shape of the map relief. With a map where the isolines have large length, their shape does not contain straight lines from an eight-neighbours area, and the number of isolines on each slice along with the number of slices will be large, then the amount of data to store the vectorized map may exceed the one of the raster map.

C. Experiment 2

1) Experiment statement: Examine the dependence of the vector map construction time on the size of the original raster grid and the number of height slices.

2) Experimental results: The map vectorization time depends quadratically on the linear map size and linearly on the number of height slices. This time is determined by the quadratic estimate of the complexity of the isolines search algorithm. The linear dependence of time on the number of slices is explained by the fact that we use an algorithm for finding contours on a plane to process slices. The results of the experiment are shown in Fig. 8.

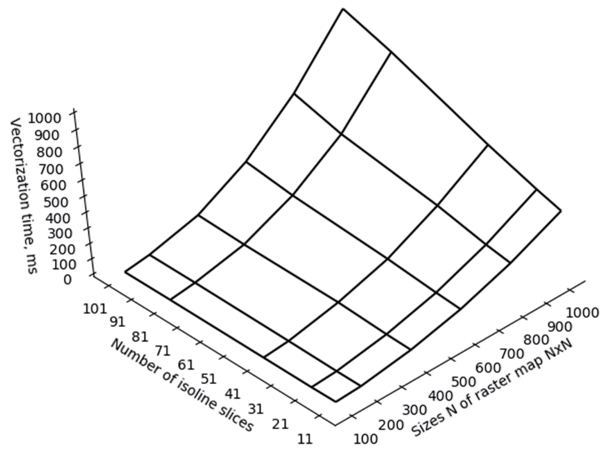

Fig. 8. Graph shows the dependence of the vectorization time on the number of slices and the size of the raster map

D. Experiment 3

1) Experiment statement: Estimate the dependence of the distribution of the error value for approximating the height of regular grid points on the number of height slices.

2) Experimental results: We found that the error values are distributed according to the normal law. The standard deviation depends exponentially on the number of slices, but does not depend on the linear dimensions of the map. For experiments, the same height interval was used, so increasing the number of slices automatically led to a decrease in the height between neighboring slices. Accordingly, the approximation error decreases exponentially when the step between the height slices decreases. The results of the experiment are shown in Fig. 9.

V. CONCLUSION

This paper describes four algorithms which preform formation of irregular grid of heights, recursively finding isolines, building and optimizing theirs contours. These algorithms can be used for raster map vectorization. Vectorized map is presented by slices contain isolines at specific height for irregular grid of heights. Isolines on each slice are n-ary tree which have inner isolines as subtrees. Oddness of nested depths level defines type of isoline either it is surface or water. The proposed format for storing map

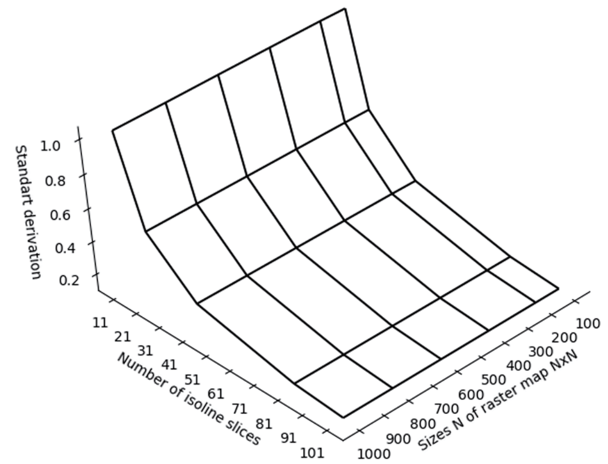

Fig. 9. Graph shows the dependence of the standard deviation on the number of slices and the size of the raster map

data will be used to represent realistic landscapes in the tasks of building ship routes.

Study found that vectorized map stored volume decries up to 2 times in compassion with raster map volume. Increase the raster size leads to quadratic build time growth and increase the number of layers leads to linear build time growth. However, vectorized map build time has never exceeded 1 second for 1 million cells map. Height discretization leads to approximation errors. Study of error distribution has demonstrated that error exponentially reduces as the height interval decreases.

Developed algorithms are proposed to be used for representation of surface model with water and ice environment.

Acknowledgment

The reported study was funded by RFBR according to the research project №18-29-03250 mk "Robust methods of synthesis of intelligent transport systems for cyber-physical objects coalition based on the Bayesian concept of probability and the modal logics".

References

[1] V. Chantsev, Y. Gudoshnikov, D. Pleshanov, A. Skutin, and A. Danshina, "Multifunctional complex model of a hydrologic system in the gulf of ob," *Vesti Gazovoy Nauki*, vol. 36, no. 4, pp. 139–148, 2018.

[2] D. Motorin and S. Popov, "Multi-criteria path planning algorithm for a robot on a multilayer map," *Information and Control Systems*, no. 3, pp. 45–53, 2018.

[3] M. Chuvatov, V. Glazunov, L. Kurochkin, and S. Popov, "The technology of management of data about wireless networks for vehicle's telematics map." in *VEHITS*, 2016, pp. 138–143.

[4] S. Nikiforov, S. Koshel, V. Frol, O. Popov, and O. Levchenko, "On the methods for the construction of seabed digital elevation models (using the example of the white sea)," *Oceanology*, vol. 55, pp. 296–305, 04 2015.

[5] V. Davidovich and A. Gangopadhyay, "An algorithm for the automatic recognition of oceanic features in frontal maps," *Oceanology*, vol. 52, no. 3, pp. 436–446, 2012.

[6] W. Li, X. Meng, Z. Wang, W. Fang, J. Zou, H. Li, T. Sun, and J. Liang, "Low-cost vector map assisted navigation strategy for autonomous vehicle," in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. IEEE, 2018, pp. 536–539.

[7] Y. G. Firsov, "Modern bathymetric survey in the arctic ocean in the context of the extended continental shelf determination in arctic (in russian)," *Vestnik gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova*, no. 6 (40), 2016.

[8] Y. Yin, H. Zhao, F. Yang, D. Shang, and J. Liu, "A vectorization method of building edge based on high resolution dsm data," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2016, pp. 1572–1574.

[9] O. Tasar, E. Maggiori, P. Alliez, and Y. Tarabalka, "Polygonization of binary classification maps using mesh approximation with right angle regularity," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 6404–6407.

[10] V. Lacroix, "Raster-to-vector conversion: problems and tools towards a solution a map segmentation application," in *2009 Seventh International Conference on Advances in Pattern Recognition*. IEEE, 2009, pp. 318–321.

[11] X. Hu, Z. Shen, J. Luo, and X. Liegang, "An adaptive parallel vectorization method for rs segmented raster map," in *The 2nd International Conference on Information Science and Engineering*. IEEE, 2010, pp. 3514–3517.

[12] L. Huarong and Z. Yi, "Vectorization of linear features on color scanned map," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, vol. 2. IEEE, 2011, pp. 1148–1152.

[13] H. Zheng, X. Zhou, and H. Wang, "Automatic vectorization of contour lines based on deformable model with adaptive flow orientation," in *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003*, vol. 2. IEEE, 2003, pp. 1151–1156.

[14] V. Bucha, S. Ablameyko, and S. Uchida, "Image pixel force fields and their application for color map vectorisation," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2. IEEE, 2007, pp. 1228–1242.

[15] S. S. Gusev, "Vectorization of raster images (in russian)," *APPLIED MATHEMATICS AND CONTROL SCIENCES*, no. 4 (40), 2018.

[16] J. Teng, F. Wang, and Y. Liu, "An efficient algorithm for raster-to-vector data conversion," *Geographic Information Sciences*, vol. 14, no. 1, pp. 54–62, 2008.

[17] M.-L. Yin, J. Chen, and Z. He, "Algorithm of drawing isoline based on improved delaunay triangle net," in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2017, pp. 1022–1026.