Nonlinear Dynamics and Neuroscience

Izvestiya Vysshikh Uchebnykh Zavedeniy. Applied Nonlinear Dynamics. 2026;34(2)

Article DOI: 10.18500/0869-6632-003204

# Study of a two-threshold modification of a biomorphic navigation system

*Y. A. Malichev*[1] ✉, *S. A. Lobov*[1,2,3], *V. G. Yakhno*[1,2]

[1]A. V. Gaponov-Grekhov Institute of Applied Physics RAS, Nizhny Novgorod, Russia
[2]Lobachevsky State University of Nizhny Novgorod, Russia
[3]Moscow Institute of Physics and Technology, Russia
E-mail: ✉u.malishev@ipfran.ru, lobov@neuro.nnov.ru, yakhno@ipfran.ru

*Received 11.10.2025, accepted 25.11.2025, available online 9.12.2025, published 31.03.2026*

***Abstract.*** The *purpose* of this work is to implement and study the dynamics of a modified version of a biomorphic visual navigation system. *Methods.* The paper uses the RatSLAM simultaneous navigation and mapping system. The RatSLAM system is a biomorphic model of visual navigation in the rodent hippocampus. In this study, we investigate a modified version of the RatSLAM system in which visual landmarks are processed using a two-threshold algorithm. *Results.* This article presents a modified version of the visual navigation system. Using a two-threshold algorithm for determining visual landmarks allows for a reduction in the size of the resulting map without loss of accuracy. Using the constructed system, location estimates and clustering metrics for visual landmarks were obtained on publicly available datasets. *Conclusion.* The constructed visual navigation system provides an estimate of the location of an object (video camera) in space that is in good agreement with the true location data. Using a two-threshold algorithm, the map size can be reduced without increasing map accuracy.

***Keywords***: simultaneous localization and mapping systems, integration paths, landmarks, neural networks, clustering.

***Acknowledgements***. This work was supported by the Ministry of Science and Higher Education of the Russian Federation under a state assignment (Project Nos. FFUF-2024-0037 — system development, FSMG-2024-0047 — research of clustering).

***For citation***: Malichev YA, Lobov SA, Yakhno VG. Study of a two-threshold modification of a biomorphic navigation system. Izvestiya VUZ. Applied Nonlinear Dynamics. 2026;34(2):314–330. DOI: 10.18500/0869-6632-003204

## Introduction

To study the navigation of living systems, it is used, among other things, the construction and study of their computational models. One of such models is RatSLAM [1, 2], based on the ideas of the navigation processes in the rodent brain. This system is a biologically relevant algorithm, for which biological compliance is more important than metric accuracy. Such models allow to significantly reduce the requirements for computational resources, especially when using neuromorphic hardware computing modules [3]. Also, an advantage of biologically relevant models is the ease of adding and changing the modalities of the system [4].

One of the main elements of the SLAM algorithms (simultaneous localization determination and mapping) and the RatSLAM algorithm is a loop closure mechanism that allows you to reset the accumulated error of path integration when hitting a known point in space. This mechanism is related to the task of place recognition, that is, recognition of the current location as one of the known ones. Recognition is performed based on data from various sensors installed on the object. The image from the video camera is often used as input data.

Traditionally, this task is approached as a search in a database of images that are closest in space to the requested one. The key issue is to create a description of the image so that images with similar descriptions are obtained at close points in space, and vice versa.

It is possible to use various types of image descriptions, such as pixel image data directly, sets of key points in the image [5], output values of artificial neural networks of various architectures [6], as well as combinations of these methods [7]. The RatSLAM algorithm uses pixel data from a compressed image as a description of images. However, for all types of descriptions, with an increase in the known area of space, the number of images in the database increases, which leads to an increase in the required amount of memory and computing resources.

In the RatSLAM algorithm, this problem is solved by storing only the most significant images in the database. This algorithm does not depend on the source of the image descriptions and can be used with more modern descriptions such as NetVLAD [6].

The quantization of space for the recognition of individual areas was also carried out in [8], however, images were classified according to known classes of locations, and clustering is performed in RatSLAM during the system operation.

However, it is possible to further improve the size of the map and the stability of the RatSLAM system by optimizing the visual landmark recognition module and reducing the number of visual landmarks created on the trajectory. This work is a development of research in [9].

## 1. Technical implementation of the visual navigation system

The system consists of three main modules: pose cells, local view cells and an experience map, and it is also possible to add a fourth module, visual odometry (Fig. 1). The visual odometry module takes images of the input video stream as input and determines the speed of the robot's movement in space. The pose cell module performs path integration and loop closure. The experience map module stores a graph of known unique points of space in a metric space. The visual landmark module maintains a collection of visual landmarks and performs their recognition.

In this paper, the main focus is on the visual landmark module.

**1.1. The visual landmark module.** The visual landmark module takes as input the image data of the input video stream. The output of the module is the current recognized visual landmark.

Visual landmarks are an expandable array of blocks, each of which corresponds to a separate visual scene in the environment. When a new visual scene appears, a new visual landmark is created, which is associated with the pixel data of that scene.

After pre-processing the video stream image (removing non-informative areas of the frame, normalizing luminances, converting to monochrome, and compressing), the visual landmark module compares the resulting visual pattern representing the current camera image with all previously learned visual landmark patterns. To do this, a similarity measure based on the sum of absolute differences (SAD) is calculated

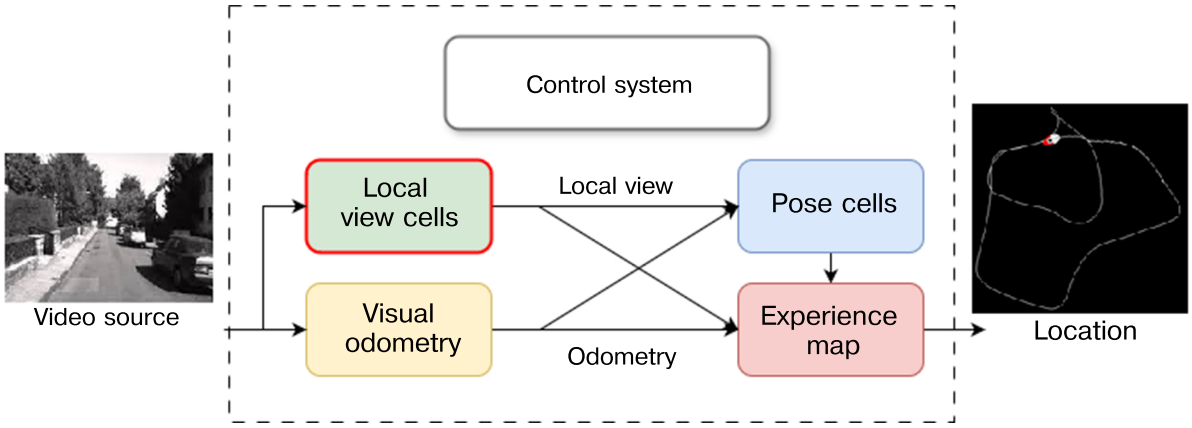


Fig. 1. Structure of OpenRatSLAM [2]. If odometry is already provided by the dataset or robot, the Visual Odometry node is not required

between the current visual pattern $p$ and each previously learned visual landmark pattern $p^j$:

$$d\left(p, p^j\right) = \frac{1}{s}\left\|p, p^j\right\|_1 = \frac{1}{s}\sum_{i=0}^{s-1}\left|p_x, {p_x}^j\right|, \tag{1}$$

where $s$ is the size of visual pattern. Next, the closest visual landmark pattern is determined:

$$b = \underset{0 \leqslant j < n}{\arg\min}\, d\left(j\right), \tag{2}$$

where $n$ is the number of known visual patterns. If the smallest difference is less than the threshold value $s_{th}$, then the corresponding existing pattern is selected ($m = 1$). Otherwise ($m = 0$), the current visual pattern is added to the pattern database

$$m = \begin{cases} 1, & d\left(b\right) \leqslant s_{th}, \\ 0, & d\left(b\right) > s_{th}. \end{cases} \tag{3}$$

The visual landmark module compares the current view with all saved view patterns to find the best match. The comparison is made when the current view and the visual landmark patterns are shifted relative to each other. The result is the current active visual landmark.

For forward-facing cameras, the comparison process allows you to cope with small rotational shifts by finding the minimum SAD when shifting the saved patterns relative to the current view in the horizontal direction by a given amount.

**1.2. Local view clustering algorithm.** The visual landmark module contains an expandable collection of images that serve as visual landmark patterns. A new visual landmark is created if the input image significantly differs from all known landmarks in terms of the metric value. Thus, we can consider this module as performing the task of clustering input images in the space of all pixel intensities. This clustering method can be classified as a hierarchical continuous clustering method with a fixed threshold [10]. The centers of the clusters are the visual landmarks, and the distance between the cluster centers and the size (radius) of the cluster correspond to the threshold value of the image similarity metric. Recognizing an input image as one of the visual landmarks is equivalent to placing a point in the pixel intensity space corresponding to the given image in the corresponding visual landmark cluster. Since the SAD (sum of absolute differences in pixel intensities) is used as a measure, this corresponds to the Manhattan distance [11] in the pixel intensity space. The Manhattan distance $d_1$ between two vectors $\mathbf{p}, \mathbf{q}$ in an $n$-dimensional real vector space with a given coordinate system is defined as the sum of the

lengths of the projections of the segment between the points onto the coordinate axes

$$d_1(\mathbf{p},\mathbf{q}) = \|\mathbf{p},\mathbf{q}\|_1 = \sum_{i=1}^{N} |p_i - q_i|. \tag{4}$$

Thus, the clusters have the shape of $n$-rhombuses. In the case where the size (radius) of the cluster is equal to the distance between the cluster centers, there is a significant overlap of the clusters. However, since the algorithm for determining visual landmarks is executed sequentially for all local views in chronological order, and the search is stopped when any of the visual landmarks is encountered in a cluster, when an input image falls into an area belonging to multiple clusters, it is recognized as the chronologically first visual landmark in the cluster where the image is located. Thus, the clusters of later visual landmarks are multidimensional rhombuses around the local viewpoint, minus the space occupied by earlier clusters. As a result, the clusters have different sizes: the first cluster and those that are sufficiently distant from each other are full-fledged multidimensional rhombuses, while all other clusters are «additions» on their borders. Additionally, the introduction of a distinction between the size (radius) of the cluster, which corresponds to the threshold for recognizing visual landmarks, and the distance between clusters, which corresponds to the threshold for creating new visual landmarks, should ensure more uniform clustering when the recognition threshold is lower than the creation threshold.

**1.3. Two-threshold algorithm for detecting visual landmarks.** The modification relates to the landmark detection module. This module supports a collection of visual landmarks, landmark images. When a new image is received from the video stream, the system decides whether to create a new landmark or to recognize one of the familiar landmarks (from the collection). The decision is based on the value of the discrepancy metric between the new image and all the images in the collection. If the smallest difference in the metric value is less than the threshold value $s_{th}$, then the corresponding existing pattern is selected ($m = 1$). Otherwise ($m = 0$), the current visual pattern is added to the pattern database according to (1), (2), (3). The previously proposed system makes a decision to create a new landmark and to recognize a known landmark based on exceeding the same threshold for the discrepancy metric. This means that when a new image is presented, the system either recognizes it as one of the known landmarks or identifies it as a new landmark. However, this approach has a drawback in that it may result in a constant jump between two landmarks when the input image is located at the boundary of the neighboring landmarks' clusters. This can lead to false identification of landmarks and, consequently, the location. Therefore, it was proposed to separate the processes of landmark creation and recognition. The creation of landmarks is performed using a higher threshold value for the discrepancy metric, $s_c$, which results in better-separated clusters (more distinct landmarks). Recognition is performed using a lower threshold value, $s_m$, which means that only images that are truly similar to a landmark are recognized as that landmark. This creates areas at the boundaries of the clusters where no landmarks are recognized. In other words, when a new image is received, the system can either recognize it as an existing landmark ($m = 1$), store it as a new landmark ($m = 0$), or do nothing ($m = -1$)

$$m = \begin{cases} 1 & d(b) \leqslant s_m, \\ -1 & s_m < d(b) < s_c, \\ 0 & d(b) \geqslant s_c. \end{cases} \tag{5}$$

This approach significantly increases the thresholds for creating new landmarks and, consequently, reduces their number by several times without significantly compromising the accuracy of the map. As a result, it reduces the memory and computational requirements of the system.

Fig. 2 illustrates this clustering mechanism by showing the results of visual landmark clustering in the «Kitti-00» video stream over a period of 120 seconds. For ease of visualization, the landmark resolution was set to 3x1, which means that there are three points in the width and one in the height. In this case, the feature space becomes two-dimensional due to the normalization of pixel intensity values

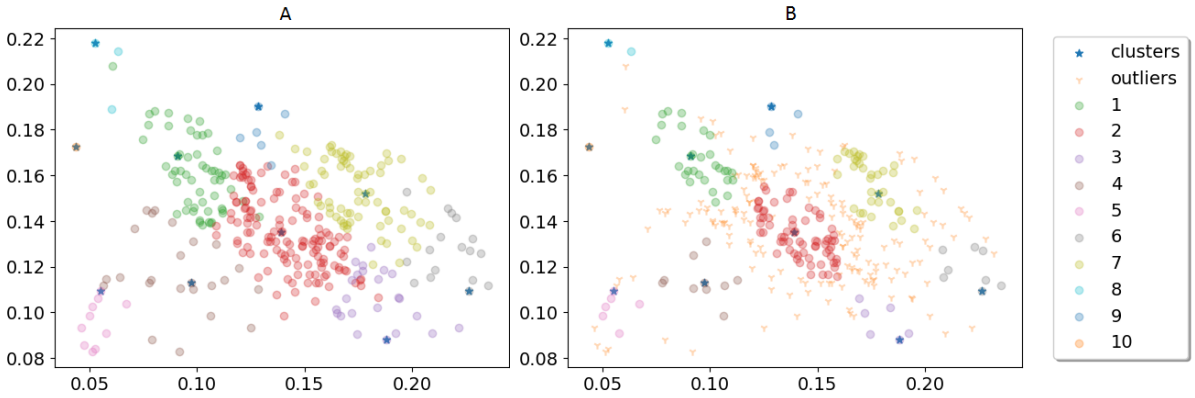


Fig. 2. Clustering of visual landmarks. (A) original algorithm ($s_{th} = 0.09$), (B) modified algorithm ($s_c = 0.09, s_m = 0.04$). Cluster centroids are marked with (*), and outliers with (+) (color online)

## 2. Testing of the developed system

To test SLAM systems, sensor data sets are used that were obtained from objects moving in space. The systems take sensor data as input and build their own estimate of the object's motion trajectory and the map of the area, and then the estimate of the trajectory obtained as a result of the system's work is compared with the true trajectory.

As a metric for comparing trajectories, in this work the average absolute error of location determination is used [12]. The absolute trajectory error (ATE) is estimated by comparing the absolute distances between the calculated and true trajectory.

Monocular visual odometry is used in the work. In this case, the dimensions of the resulting trajectory are made in a relative scale.

**2.1. Clustering estimation methods.** To evaluate the quality of clustering, this paper uses metrics such as the number of objects in a cluster, the intercluster distance, the cluster diameter, the distance between the centroids of the clusters, and the Dunn index metric [13]. The Manhattan distance is also used as a distance metric.

The intercluster distance $\delta$ (separation score) is defined as the minimum distance between a point belonging to a given cluster and a point in another cluster:

$$\delta(c_k, c_l) = \min_{x_i \in c_k, x_j \in c_l} \|x_i - x_j\| . \tag{6}$$

The cluster diameter $\Delta(c_k)$ (cohesion score) is defined as the maximum distance between points in the same cluster:

$$\Delta(c_k) = \max_{x_i, x_j \in c_k} \|x_i - x_j\| . \tag{7}$$

Since the centroids of the clusters are fixed in the clustering method used, and do not change since the cluster was formed, it is advisable to use the radius of the cluster, the maximum distance from the cluster centroid to an object belonging to the given cluster, as a measure of cohesion:

$$\Delta_1(c_k) = \max_{x_i \in c_k} \|x_i - c_k\| . \tag{8}$$

The distance between clusters is defined as the distance between the cluster centroids:

$$\delta^4(c_k, c_l) = \|\overline{c_k} - \overline{c_l}\| . \tag{9}$$

A modified version of the Dunn index [14] is chosen as a metric [14],

$$D(C) = \frac{\min_{c_k \in C}\{\min_{c_l \in C \setminus c_k}\{\delta(c_k, c_l)\}\}}{\max_{c_k \in C}\{\Delta_1(c_k)\}}. \tag{10}$$

Higher values of the Dunn index correspond to better clustering.

**2.2. Data sets used for testing.** This work uses the public KITTI dataset, sequences 00 and 02 [15], as well as the TUM RGB-D dataset, sequence fr2/pioneer_slam [12]. The datasets contain video stream frames and true trajectories.

## The results and their discussion

**2.3. Trajectory detection metrics.** When the system was tested on the datasets used, the metrics presented in Table 1 were obtained.

Fig. 3 shows the distribution of the average absolute localization error (normalized by the maximum error, shown in color) on the corresponding trajectory for different thresholds for creating (Y axis) and recognizing (X axis) landmarks.

The main diagonal represents the case of the original system, in which the thresholds are equal. We can see that at high threshold the error increases sharply due to incorrect landmark recognizing. However, as we move away from the main diagonal towards lower abscissa (where the recognizing threshold is lower than the creation threshold), small error values are restored.

Fig. 4 shows a graph of the main diagonal of Fig. 3 (blue graph, corresponding to the original

Table 1. Model performance metrics (m)

| Sequence | Metrics | Parameters | System | |
|---|---|---|---|---|
| | | | RatSLAM | |
| | | | One threshold | Two threshold |
| Kitti-00<br>one threshold:<br>$s_{th} = 0.075$<br>two threshold:<br>$s_{th} = 0.15$<br>$s_m = 0.085$ | ATE | Max | 37.487 | 44.949 |
| | | Mean | 18.765 | 18.266 |
| | | Median | 19.162 | 16.389 |
| | Number of visual landmarks | | 2456 | 646 |
| | Map size, MB | | 4.73 | 1.01 |
| Kitti-02<br>one threshold:<br>$s_{th} = 0.075$<br>two threshold:<br>$s_{th} = 0.15$<br>$s_m = 0.085$ | ATE | Max | 122.099 | 121.101 |
| | | Mean | 24.279 | 25.489 |
| | | Median | 22.234 | 24.293 |
| | Number of visual landmarks | | 2503 | 556 |
| | Map size, MB | | 5.36 | 2.03 |
| TUM fr2/pioneer_slam<br>one threshold:<br>$s_{th} = 0.03$<br>two threshold:<br>$s_{th} = 0.05$<br>$s_m = 0.03$ | ATE | Max | 0.734 | 0.783 |
| | | Mean | 0.385 | 0.393 |
| | | Median | 0.382 | 0.354 |
| | Number of visual landmarks | | 326 | 149 |
| | Map size, kB | | 861.3 | 406.1 |

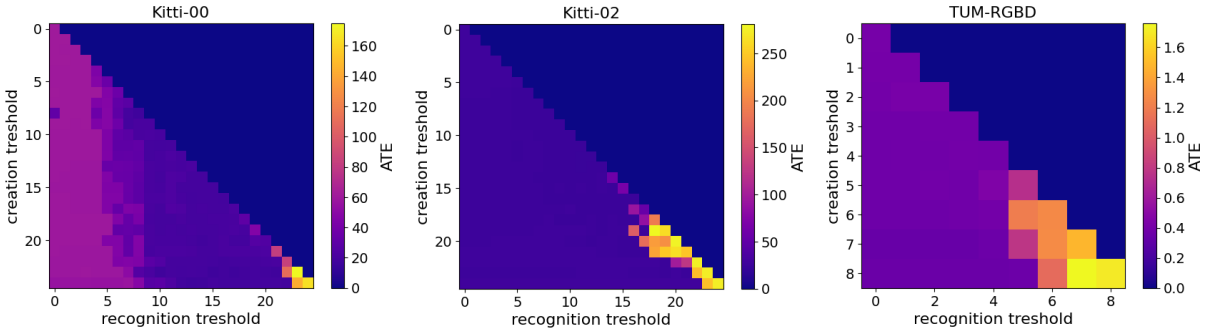


Fig. 3. Average absolute trajectory error distribution on trajectories for different thresholds for creating $s_c$ (Y axis) and recognizing $s_m$ (X axis) landmarks (color online)

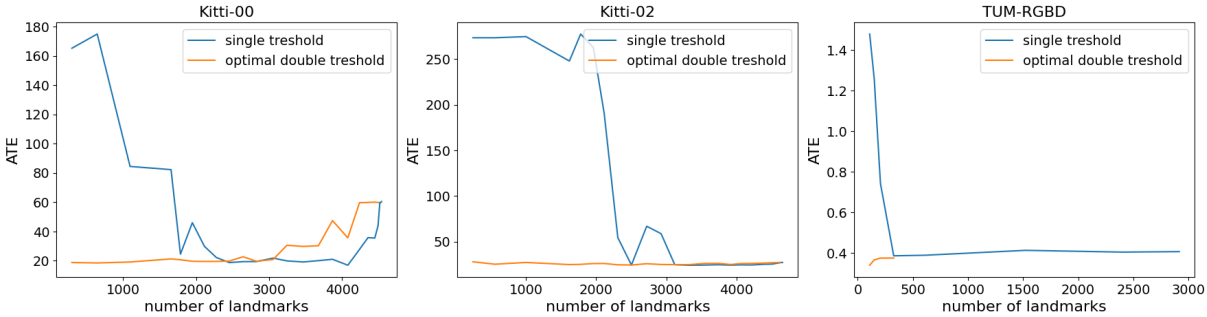


Fig. 4. Dependence of the average absolute localization error on the number of stored visual landmarks on the trajectories (color online)

system), as well as a graph shifted 4 values to the left of the main diagonal (orange graph), but with the number of landmarks created on the trajectory on the X axis.

The ordinate axis represents the magnitude of the average absolute localization error. For the original system, the error on the Kitti-00 trajectory begins to grow when the number of landmarks is less than 2,000. For the system with two thresholds, the error begins to grow when the number of landmarks is less than 300.

The following are visualizations of the trajectories obtained during testing. The dotted lines represent the true trajectories, while the colored lines represent the estimated trajectories generated by the system. The color represents the magnitude of the deviation between the estimated and true trajectories.

Fig. 5-I shows an example of the system's operation with identical low thresholds (0.01). In this case, a unique landmark is created for almost every image, resulting in almost no loop closures (changes in location when identifying known landmarks). The error value corresponds to velocity integration (odometry).

Fig. 5-II shows an example of the system operating with the same critical thresholds (0.075). The ideal case for the original system is that all possible loop closures are performed.

Fig. 5-III shows an example of the system's operation with identical critical thresholds (0.09). When the threshold is raised above the critical value, false loop closures occur, and the error value begins to increase. As the threshold is further increased, the error value rapidly increases, and the topological similarity between the map and the real space is lost.

Fig. 5-IV shows a graph for an ideal option in a two-threshold system. The threshold for creating landmarks is 0.15 (more than the one at which the map for the original system «deteriorates»), the recognition threshold is 0.085. The error value differs slightly from the ideal single-threshold case, however, the number of necessary landmarks is several times less.

In the case of an excessively low threshold for recognizing landmarks, landmarks are rare in a two-threshold system, so no loops are closed, and the map corresponds to trajectory integration.

The main feature of the Kitti-02 and TUM datasets is the low number of possible loop closures and the high quality of odometry without the use of visual landmarks. Due to this, the effect of loop closures on the error value is small. However, there remains an increase in location errors as the thresholds for recognizing visual landmarks increase, caused by erroneous loop closures. Thus, at low thresholds, the error value is almost constant.

**2.4. Clustering metrics.** When testing the system on the KITTI-00 dataset, clustering metrics for visual landmarks were obtained. For a system with one threshold, the threshold was set $s_{th} = 0.15$, and for the two-threshold system, the threshold for creating landmarks was set to $s_c = 0.15$, and the threshold for recognition was set to $s_m = 0.085$. Fig. 6 shows the histograms of the distribution of clusters based on the number of visual landmarks they contain. On the right, the results are shown for the original single-threshold system, and on the left, the results are shown for the modified two-threshold system. From the figure, it can be seen that for the single-threshold system, the total number of objects in the clusters, as well as the average number of objects in a single cluster, is higher than in the two-threshold

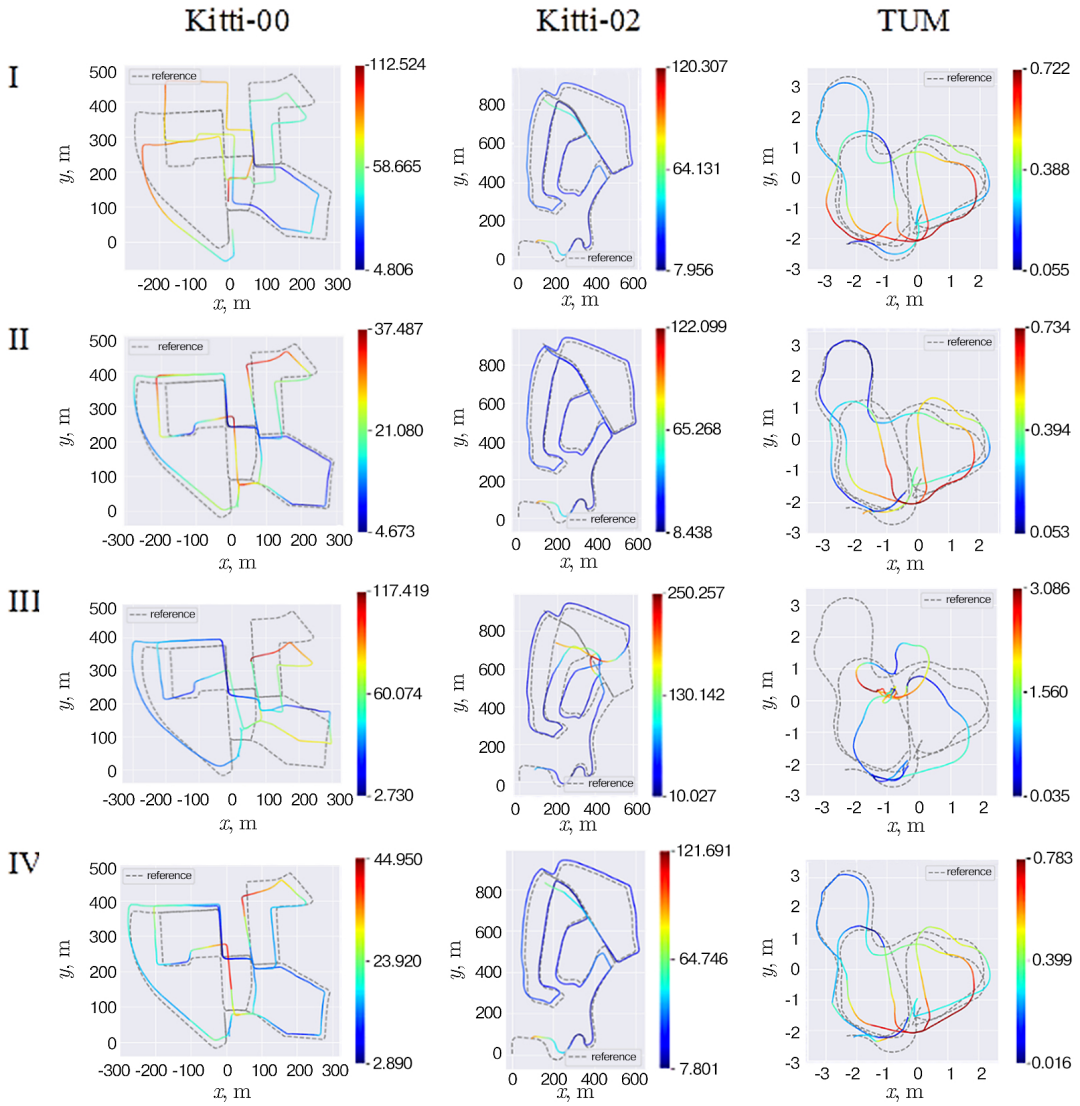


Fig. 5. Visualizations of the trajectories of the original and modified systems on the Kitti 00.02 and TUM trajectories. Graphs of the distribution of the average absolute localization error for different values of the landmark creation and recognition thresholds are shown. $s_c$ is the landmark creation threshold, $s_m$ is the recognition threshold. Kitti: (I) $s_c = 0.01$, $s_m = 0.01$ (II) $s_c = 0.075$, $s_m = 0.075$ (III) $s_c = 0.09$, $s_m = 0.09$ (IV) $s_c = 0.15$, $s_m = 0.08$. TUM: (I) $s_c = 0.01$, $s_m = 0.01$ (II) $s_c = 0.03$, $s_m = 0.03$ (III) $s_c = 0.05$, $s_m = 0.05$ (IV) $s_c = 0.05$, $s_m = 0.03$ (color online)

system. This is because in the two-threshold system, there are visual landmarks that do not belong to any cluster, which results in a decrease in the total number of visual landmarks in the clusters. Additionally, in the two-threshold system, there are visual landmarks that are truly close to each other, which results in a decrease in the average number of visual landmarks in a single cluster.

The metrics of the minimum distance from the cluster centroid to the points belonging to other clusters were also obtained. The distribution of clusters by metric values is also shown in Fig. 6. From the figure, it can be seen that for a system with a single threshold, the average distance to the points of the neighboring cluster is less than the distance between the cluster centroids, which corresponds to the threshold value of $s_{th} = 0.15$. This is because in a system with a single threshold, the clusters can overlap, and the points of one cluster can almost touch the centroid of another cluster. This leads to ambiguous recognition of points in the overlapping area, which in turn leads to unstable recognition of visual landmarks.

In a system with two recognition thresholds, the distribution is shifted towards the more distant

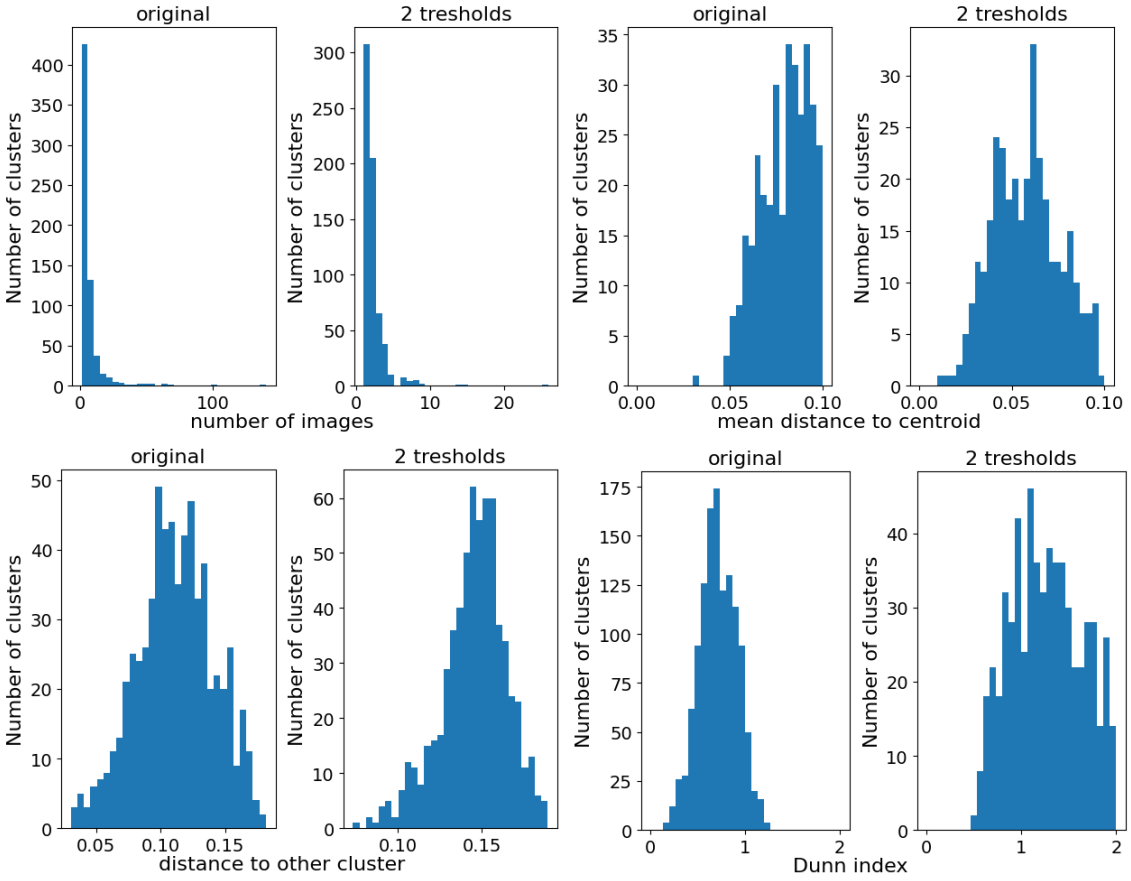


Fig. 6. Distributions of clusters in the original and two-threshold systems on the Kitti-00 data set by the number of points, by the distance to points of other clusters, by the average distance from the cluster centroid to the cluster points, and by the Dunn index value

points, and the majority of clusters have the nearest point of the neighboring cluster at a distance comparable to the intercluster distance of $s_c = 0.15$. In this case, there are also overlapping areas (distances ranging from 0.075 to 0.085), but for the vast majority of clusters, there are no points within this range.

The compactness metric of the clusters was also obtained, which is the average distance from the cluster centroid to other points in the cluster. The distribution of clusters based on the metric value is shown in Fig. 6. From the figure, it can be seen that for a single-threshold system, the average distance to the cluster centroid is approximately half the size of the distance for a two-threshold system. Therefore, the clusters in the two-threshold system are more compact.

Next, based on the metrics of cluster compactness and distance, the Dunn indices for the clusters were calculated. The distribution of clusters by metric values is shown in Fig. 6. From the figure, it can be seen that for a single-threshold system, the majority of clusters have a Dunn index less than 1, which corresponds to overlapping clusters. For a two-threshold system, the majority of clusters have a metric value greater than 1, which corresponds to compact, distant clusters. Therefore, clustering in a two-threshold system is more effective. Fig. 7 shows similar graphs for the TUM dataset. The graph data exhibits the same patterns as the Kitti-00 dataset.

The testing results of the systems are presented in Table 2.

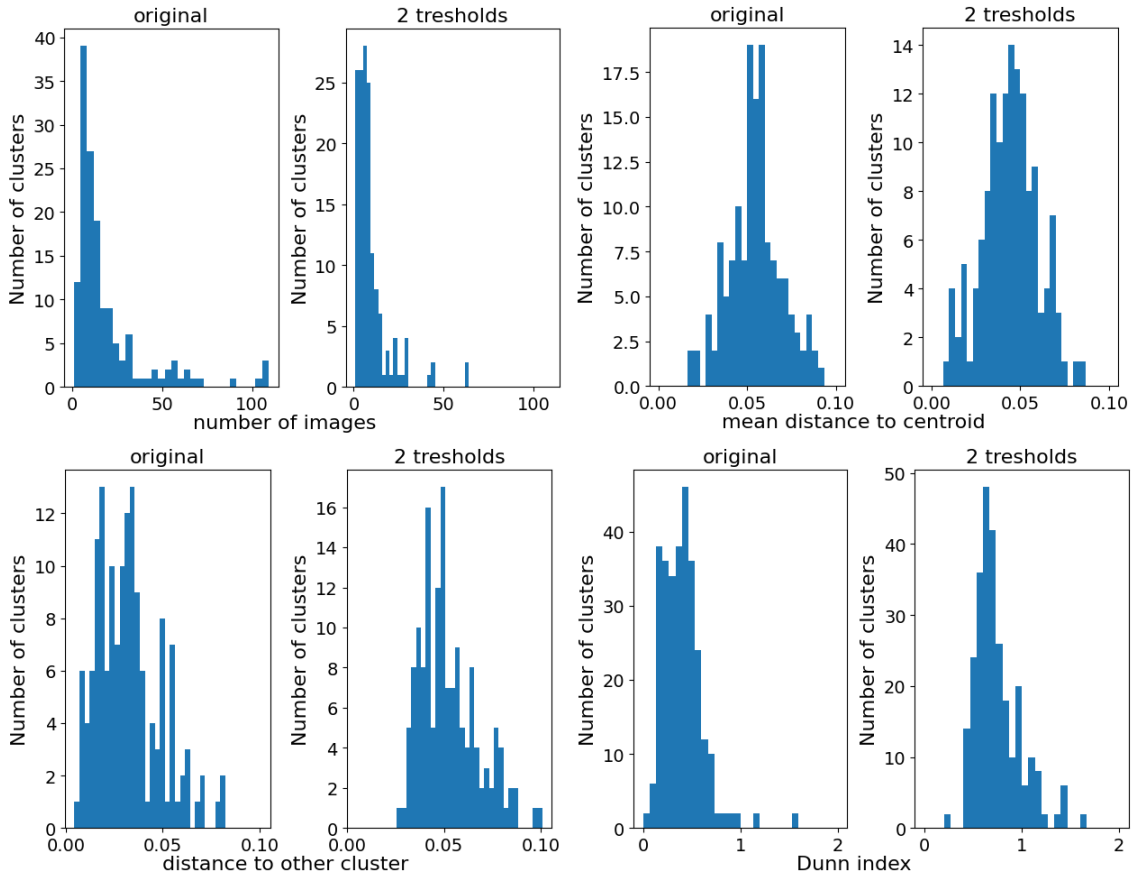


Fig. 7. Distributions of clusters in the original and two-threshold systems on the TUM data set by the number of points, by the distance to points of other clusters, by the average distance from the cluster centroid to the cluster points, and by the Dunn index value

Table 2. Clustering metrics

| | Metrics | System | |
|---|---|---|---|
| | | One threshold | Two threshold |
| Kitti-00 | Number of objects in the cluster | 7.01 | 2.01 |
| | Average cluster radius | 0.114 | 0.062 |
| | Minimum distance to points in the adjacent cluster | 0.111 | 0.145 |
| | The Dunn Index | 0.71 | 1.45 |
| TUM | Number of objects in the cluster | 19.45 | 9.84 |
| | Average cluster radius | 0.053 | 0.045 |
| | Minimum distance to points in the adjacent cluster | 0.032 | 0.052 |
| | The Dunn Index | 0.40 | 0.81 |

## 3. Discussion

The conducted research is devoted to the development and comprehensive analysis of a modified two-threshold algorithm for the RatSLAM biomimetic navigation system. The main goal of the work was to optimize the process of visual landmark clustering in order to reduce the computational load and

memory consumption without significantly compromising localization accuracy.

The results of the experimental evaluation on standard datasets have convincingly demonstrated the effectiveness of the proposed approach. A key achievement is the demonstration that dividing the overall visual landmark recognition threshold into two independent thresholds, the threshold for creating new landmarks and the threshold for recognizing them, allows for a significant reduction in the overall number of visual patterns generated. This reduction directly leads to a significant decrease in the final size of the map of the area. At the same time, as shown by the absolute trajectory error (ATE) metrics, the accuracy of the system's location estimation remains comparable to the original algorithm, and in some scenarios, it even improves due to a decrease in the number of false positives during loop closure.

A deep analysis of clustering using compactness metrics (average cluster radius), separability (minimum distance to the neighboring cluster), and the generalized Dunn index revealed a qualitative improvement in the data structure. In the two-threshold system, the clusters become more compact and better separated from each other, as evidenced by an increase in the Dunn index from 0.71 to 1.45 for KITTI-00 and from 0.40 to 0.81 for TUM. This indicates a more stable and consistent recognition process that minimizes the «jumping» between landmarks at the cluster boundaries.

Despite the positive results, it is important to note a few limitations and identified challenges. Firstly, the system's performance heavily relies on manually adjusting the optimal threshold values, which can vary depending on environmental characteristics (for example, texture and illumination) and sensor parameters. Automating or adaptive selection of these thresholds is an important task for future research. Secondly, the current implementation is still based on low-level pixel features (SAD), which makes it sensitive to changes in lighting, weather conditions, and dynamic objects in the frame. Thirdly, in scenarios with very limited visual information or long-term drift without loop closure, the algorithm's advantages may be offset by the accumulation of odometry errors.

There are several promising directions for further development of this work. First, it is necessary to expand the multimodality of the system by integrating data from additional sensors (lidars, IMUs), which will increase its robustness in various conditions. Second, it is advisable to replace pixel comparison with modern neural network descriptors (such as NetVLAD), which are more invariant to changes in the scene's appearance.

A key step towards moving from laboratory experiments to practical applications is the integration of the proposed algorithms into a full-fledged robotic system capable of navigating in dynamically changing environments. A promising approach to achieving this goal is to combine the optimized RatSLAM with the generalized cognitive map model proposed by the V. A. Makarov group [16, 17]. This model has demonstrated its versatility and effectiveness in both wheeled robot control tasks [17] and motion planning for robotic manipulators [18]. Its high biological relevance, which has been confirmed in psychophysiological experiments on humans [19, 20], allows us to create a truly bio-inspired hybrid architecture.

Thus, the future neuromorphic system may include the proposed two-threshold module for low-level navigation, which is connected to a higher-level decision-making core. This core could be a semantic cognitive map graph, as proposed in [21], allowing the robot to not only navigate but also understand the semantics of its environment, adaptively planning its actions in complex, unstable, and dynamic environments.

## Conclusion

A two-threshold algorithm for determining visual landmarks has been proposed and implemented. The use of this algorithm allows for reducing the required size of the space map without significant loss in the accuracy of the resulting map. The proposed algorithm has demonstrated its effectiveness.

In the future, it is planned to expand the set of signal modalities used in place recognition, as well as to develop algorithms for combining signals of different modalities to improve the stability of the system. In addition, it is planned to implement more advanced image comparison methods based on neural network algorithms and integrate these algorithms into a robotic system capable of navigating in dynamically changing environments.

## References

1. Milford MJ. Robot Navigation from Nature. Berlin: Springer; 2008. 196 p. DOI: 10.1007/978-3-540-77520-1.
2. Ball D, Heath S, Wiles J, Wyeth G, Corke P, Milford M. OpenRatSLAM: an open source brain-based SLAM system. Auton. Robot. 2013;34:149–176. DOI: 10.1007/s10514-012-9317-9.
3. Hines A, Milford M, Fischer T. A compact neuromorphic system for ultra–energy-efficient, on-device robot localization. Sci. Robot. 2025;10(103):eads3968. DOI: 10.1126/scirobotics.ads3968.
4. Milford M, Jacobson A. Brain-inspired sensor fusion for navigating robots. In: IEEE International Conference on Robotics and Automation. 2013, Karlsruhe, Germany. IEEE; 2013. P. 2906–2913. DOI: 10.1109/ICRA.2013.6630980.
5. Campos C, Elvira R, Rodríguez JJG, Montiel JMM, Tardós JD. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. IEEE Transactions on Robotics. 2021;37(6):1874–1890. DOI: 10.1109/TRO.2021.3075644.
6. Arandjelovic R, Gronat P, Torii A, Pajdla T, Sivic J. NetVLAD: CNN architecture for weakly supervised place recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, Las Vegas, NV, USA. IEEE; 2016. P. 5297–5307. DOI: 10.1109/CVPR.2016.572.
7. Cao B, Araujo A, Sim J. Unifying Deep Local and Global Features for image search. In: Vedaldi A, Bischof H, Brox T, Frahm JM, editors. European Conference on Computer Vision (ECCV). Cham: Springer; 2020. P. 726–743. DOI: 10.1007/978-3-030-58565-5_43.
8. Berton G, Trivigno G, Caputo B, Masone C. EigenPlaces: Training Viewpoint Robust Models for Visual Place Recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023, Paris, France. IEEE; 2023. P. 11046–11056. DOI: 10.1109/ICCV51070.2023.01017.
9. Malishev YA, Yakhno VG. Biomorphic navigation system version. Izvestiya VUZ. Applied Nonlinear Dynamics. 2024;32(5):606–624. DOI: 10.18500/0869-6632-003107.
10. Mousavi M, Bakar AA, Vakilian M. Data stream clustering algorithms: A review. Int. J. Adv. Soft Comput. Appl. 2015;7:1–15.
11. Yadav DP, Kumar NK, Sahani SK. Distance metrics for machine learning and it's relation with other distances. Mikailalsys J. of Math. and Statistics. 2023;1(1):15–23. DOI: 10.58578/ mjms.v1i1.1990.
12. Sturm J, Engelhard N, Endres F, Burgard W, Cremers D. A benchmark for the evaluation of RGB-D SLAM systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2012. P. 573–580. DOI: 10.1109/IROS.2012.6385773.
13. Dunn JC. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics. 1973;3(3):32–57. DOI: 10.1080/01969727308546046.
14. Bezdek JC, Pal NR. Cluster validation with generalized Dunn's indices. In: Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems. IEEE; 1995. P. 190–193. DOI: 10.1037/h0061626.
15. Geiger A, Lenz P, Stiller C, Urtasun R. Vision meets robotics: The kitti dataset. The International Journal of Robotics Research. 2013;32(11):1231–1237. DOI: 10.1177/0278364913491297.
16. Villacorta-Atienza JA, Velarde MG, Makarov VA. Compact internal representation of dynamic situations: neural network implementing the causality principle. Biol. Cybern. 2010;103(4):285–297. DOI: 10.1007/s00422-010-0398-2.
17. Villacorta-Atienza JA, Makarov VA. Neural network architecture for cognitive navigation in dynamic environments. IEEE Transactions on Neural Networks and Learning Systems. 2013;24(12):2075–2087. DOI: 10.1109/TNNLS.2013.2271645.
18. Villacorta-Atienza JA, Calvo C, Lobov S, Makarov VA. Limb movement in dynamic situations based on generalized cognitive maps. Mathematical Modelling of Natural Phenomena. 2017;12(4):15–29. DOI: 10.1051/mmnp/201712403.
19. Villacorta-Atienza JA, Calvo C, Díez-Hermano S, Sánchez-Jiménez A, Lobov S, Krilova N, Murciano A, López-Tolsa GE, Pellón R, Makarov VA. Static internal representation of dynamic situations reveals time compaction in human cognition. Journal of Advanced Research. 2021;28: 111–125.

DOI: 10.1016/j.jare.2020.08.008.
20. Lobov SA, Krilova NP, Makarov VA, Kurganov DP, Makarova J. Arcade game testing of generalized cognitive maps in humans. In: 2021 Third International Conference Neurotechnologies and Neurointerfaces (CNN). 2021, Kaliningrad, Russian Federation. IEEE; 2021. P. 61–63. DOI: 10.1109/CNN53494.2021.9580220.
21. Calvo C, Villacorta-Atienza JA, Díez-Hermano S, Khoruzhko M, Lobov S, Potapov I, Sánchez-Jiménez A, Makarov VA. Semantic knowledge representation for strategic interactions in dynamic situations. Front. Neurorobot. 2020;14:4. DOI: 10.3389/fnbot.2020.00004.