Merging occupancy grid map based on Transferable Belief Model

Dobrokhvalov Maxim
Saint-Petersburg Electrotechnical University "LETI"
Saint-Petersburg, Russia
Email: modobrokhvalov@gmail.com

Anton Filatov
Saint-Petersburg Electrotechnical University "LETI"
Saint-Petersburg, Russia
Email: ant.filatov@gmail.com

***Abstract*—This paper presents a new algorithm for merging occupancy grid maps, the cells of which are based on the Transferable Belief Model. The developed algorithm excludes the collision of the merged maps. There are no disjoint areas in the source maps in the final map. In addition, this article explores possible modifications of the ORB algorithm descriptor using the Transferable Belief Model. Testing was conducted using the MIT Stata center dataset. The source code is available on https://github.com/Nightbot1448/TBMMapMerging.**

***Index Terms*—occupancy grid map, map merging, Transferable Belief Model, descriptor of keypoint**

I. Introduction

Occupancy grid maps are one of the variants of map representation formats that are built by robots in the process of performing the task of simultaneous localization and mapping[1]. The use of such maps is advisable when there is no a priori information about the environment. Such maps are used by autopilots, rescue robots and other robots that use information about the space around them to move. Building a map using data from only one robot is slow. Also there are a lot of errors that are difficult to fix. These disadvantages can be overcome by using more than one robot, and the maps they build can be combined into one. The difficulty lies in correcting conflicts that arise when merging maps.

Occupancy maps often use probabilistic theory to provide information about the occupancy of a cell [2]. In this paper, it is proposed to consider a different approach to storing information about space using occupancy maps. The mathematical basis for this approach is the Dempster–Schaefer theory (DST) [3]. One of the directions of development of this theory is the Transferable Belief Model(TBM) [4].

Section II will describe the difference between classical probability theory and TBM. It will also be said about the representation of the cell in this work. The section III will provide a brief overview of some of the existing solutions. Section IV will describe the presentation of the map, the idea of the algorithm, and also talk about the modification of descriptors and the way of aligning maps. Subsection V-A describes the selection of a merge rule, subsection V-B compares various descriptor modifications, and subsection V-C describes the accuracy of the merge and the computation time of the algorithm.

II. Mathematical Base

Consider some of the basis of Dempster-Schafer theory and the Transferable Belief Model. These theories are the basis of this work. For ease of further understanding, everything will be built by analogy with the states of the map cell.

$$X = \{h, \overline{h}\} \tag{1}$$

$$p(h) + p(\overline{h}) = 1 \tag{2}$$

$$2^X = \{\emptyset, \{h\}, \{\overline{h}\}, X\} \tag{3}$$

$$\sum_{A \in 2^X} m(A) = 1 \tag{4}$$

Let h be some event (eq. (1)). In classical probabilistic theory, the sum of the probabilities of an event and complementary event is equal to 1 (eq. (2)). In TBM there is a transition to the superset (eq. (3)). Accordingly, the state is described by 4 masses, which show the measure of the fact that the cell is in this state. The sum of the masses must be equal to 1 (eq. (4)). Lack of knowledge or uncertainty between the states h and $\overline{h}$ is explicitly described by the mass of X, and if two sources provide conflicting information, then the mass of conflict $\emptyset$ increases.

Consider the projection of this theory onto the cell representation of the occupancy map. The set of states is two atoms [5] $X = \{Occupied, Free\}$. Thus, the superset is the set of four elements $2^X = \{\emptyset, \{Occupied\}, \{Free\}, \{Occupied, Free\}\} = \{Conflict, Occupied, Free, Unknown\}$. In this way $m(\emptyset) = m(Conflict)$ – the mass that the cell is in a conflict state, $m(Occupied)$ – the mass of the fact that the cell is occupied, $m(Free)$ – the mass that the cell is free, $m(\{Occupied, Free\}) = m(Unknown) = m(X)$ – remaining unallocated mass that reports a lack of information.

In this paper, the differences between DST and TBM are limited to the fact that TBM allows for the possibility that the conflict mass will be non-zero. This is prohibited in DST. DST always requires conflict normalization. Conflict normalization is the distribution of the mass of the conflict between all other possible states in proportion to their masses. However, in the implementation, the conflict is always normalized.

III. Related Work

At the moment there are many different algorithms for merging occupancy maps. Some of the existing algorithms will be surveyed below. As sources of information on the approaches to merging maps, we selected articles describing the merging of occupancy maps, differing in merging approaches for the possibility of comparison. The articles were published in 2005-2020.

Andreas Birk and Stefano Carpin in work [6] propose an algorithm based on the Adaptive random walk algorithm with heuristics. The heuristic used in the algorithm is based on a special image similarity function, calculated in linear time. To determine the quality of the merge, a special function is introduced that shows the quality of the merge of the maps:

$$ai\,(m_1, m_2) = 1 - \frac{agr(m_1, m_2)}{agr(m_1, m_2) + dis(m_1, m_2)},$$

where $agr(m_1, m_2)$ – measure of map consistency, $dis(m_1, m_2)$ – measure of inconsistency. In this work, a cell can be in one of three states - occupied, free, no information. If the cell values do not match when merged, then the cell is marked as unknown. The runtime is 170 seconds on a Pentium IV 2.2 GHz processor. The main disadvantage in this work is the discrete representation of the cell state. This is a disadvantage, since it does not express a measure of confidence in the correctness of the data. Also the disadvantage is the duration of the work. However, the merge method is an advantage, since the lack of information is of higher priority than the presence of unreliable information.

In [7] Li Hao and his colleagues describe an objective function based on the probability of being occupied in the map fusion section. Also presented are some methods that are developed on the basis of a genetic algorithm to optimize the objective function. Based on this method, a general solution for the association of several robots is the additionally described strategy for indirect estimation of the relative position of robots. The proposed method can perform the task of merging, even with a larger initial map alignment error and high inconsistency inherent in the map. The merging process is averaging the values of the corresponding cells in the original maps. The operating time is 15 seconds on a processor with a frequency of 3.0 GHz. In this work, the disadvantage is the averaging of the values of the cells of the merged maps. This is a disadvantage as the conflict remains. Among the considered analogs, this algorithm has the highest performance.

Sajad Saeedi and colleagues in their work [8] extend the Generalized Voronoi Diagram (GVD) to encapsulate probabilistic information encoded in the occupancy grid map. A new construct, called Probabilistic GVD (PGVD), works directly with occupancy grids and is used to determine the relative transformation (offset and rotation) between maps and combine them. The merging process consists of several stages. Initially, there is a relative transformation between the two maps. Then the probabilities are combined and then filtered to get the final map. The data obtained as a result of the map transformation is included using the additive property of the logarithmic representation of occupancy. An entropy filter is applied to the merged map. The entropy filter compares the merged and original maps and rejects updates that increase entropy. Mutual information is defined as the decrease in entropy in cell (i, j) between the original map and the merged map. The resulting map is defined as follows: if the value of the mutual information for the cell is non-negative, then data is taken from the merged map, otherwise - from the original one. The runtime is 34 seconds on a Core2Duo 2.66 GHz processor. In this work, the main disadvantage is the operating time. The main advantage of this work is the way of resolving the conflict when merging maps.

In paper [9] by Guillaume Trehard and colleagues, the merging algorithm is based on a Transferable Belief Model for occupancy maps. The cell representation used in this paper is described in section II. The merge is performed according to the conjunctive rule [10]. When determining map consistency, the disjunctive orthogonal operator [9] is used. The main advantage of this work is the cell presentation. Using TBM for representation broadens the view of the space. As will be shown later, this work is the foundation for the current.

Andrew Howard suggests the approach [11], which uses the Bayesian inference [12] and the particle filter [13]. The algorithm works in conjunction with the task of simultaneous localization and mapping. Merging takes into account the time-reversed sequence of maps states. The resulting cell value is calculated by calculating the average of the merged maps. The advantage is the use of a particle filter as this allows to choose the best option among the many. However, averaging the merge value is a disadvantage.

The articles do not provide a link to the code, as a result of which there is no way to test it under equal conditions - on the same processor and the same dataset. As a result, the numbers are taken from the works themselves, and information about the processors and/or their frequency is indicated for comparison.

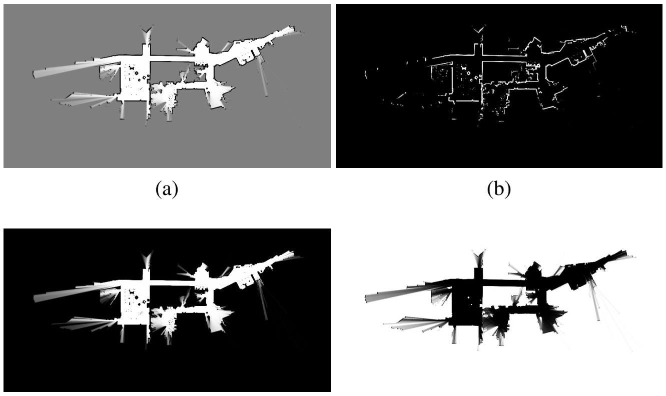
IV. Implemetation

This section describes the map representation used in the proposed occupancy grip map merging algorithm. The occupancy map is a two-dimensional grid of cells. In this paper, the cell is based on TBM. Thanks to this, it is possible to store in the cell more information about the space. The algorithm is based on images of different occupancy map representations. To select an algorithm for detecting keypoints, a comparison of various algorithms for detecting features has been performed.

A. Map representation

The TBM cell is the main feature of the occupancy grid representation considered in this algorithm. The cell of the map is similar to that considered in the [9] algorithm.

The cell representation used in this paper is described in section II. In fact, the map can be represented as 3 different maps, each of which carry some specific information about the area on the map. It is possible to get maps showing an



(a) (b) (c) (d)

Figure 1. Map representations. (a) – a probabilistic representation, (b) – a representation showing an occupied area, (c) – a representation showing an free area, (d) a representation showing an area that is unknown

occupied area, a free area, an area about which there is no information. There is possible convert a cell of type TBM to the usual probabilistic value [4] using the formula

$$P_{Bet}(x) = \sum_{x \in A \subseteq X} \frac{m(A)}{|A|},$$

where $x \in X$ are atoms, $|A|$ is the number of x atoms that appear in A, X is the set of states.

Fig. 1a provides the probabilistic representation of the occupancy map, in which an unoccupied cell corresponds to 1.0 (white), an occupied cell - 0.0 (black), a cell about which there is no information - 0.5. Different representations of the occupancy map are illustrated in fig. 1b - 1d. In these figures, white color corresponds to the fact that the entire mass of the cell is concentrated on this belief, black color - the mass of this belief is equal to 0 in this cell. As it was said, the conflict is normalized, so the mass is distributed among the three indicated beliefs. The sum of the masses of these three beliefs is 1, that is, $m\,(Occupied) + m\,(Free) + m\,(Unknown) = 1$.

B. Algorithm idea

On the probabilistic map (1a), keypoints are extracted using the ORB [14] feature detector. Then the descriptors of the feature points in the images corresponding to the three maps listed above are calculated. The resulting descriptors are concatenated. Thus, descriptors of keypoints of the map are obtained. Next, the descriptors are matched. The matched keypoints are split according to the maps to which they belong. Then the computation of the affine transformation is performed. Once a transform is found, it is applied to the map for which it was calculated. The maps are merged using the disjunctive rule, which is described below. Alg. 1 demonstrates pseudocode of the algorithm.

C. Comparison of descriptors

Initially, ORB, BRIEF [15], SURF [16], SIFT [17] were considered as variants of the used detectors and descriptors.

Algorithm 1 TBM map merging

$imgs \leftarrow$ get_imgs_from_maps()
$keypoints_1, keypoints_2 \leftarrow$ get_ORB_keyposints($imgs$)
$d_1, d_2 \leftarrow$ get_descriptors($keypoints_1, keypoints_2$)
$desc\_map_1, desc\_map_2 \leftarrow$ concatenate(d_1, d_2)
$matched \leftarrow$ match_and_filter($desc\_map_1, desc\_map_2$)
$transform \leftarrow$ compute_transform($matched$)
$transfomed \leftarrow$ apply_transform($map_2, transform$)
return merge_disjunctive($map_1, transfomed$)

Table I
COMPARISON OF DIFFERENT DETECTORS AND DESCRIPTORS ON THE MAPS IMAGES

Expiriment	ORB(1)	BRISK(2)	SIFT(3)	SURF(4)	Best
1	0.958	0.889	0.25	0.0	1
2	1.0	1.0	1.0	0.951	1, 2, 3
3	0.714	0.5	0.0	0.0	1
4	0.993	1.0	1.0	1.0	2, 3, 4
5	0.96	0.818	0.0	0.0	1
6	0.999	1.0	1.0	1.0	2, 3, 4
7	0.999	0.998	1.0	0.996	3

For comparative analysis, several maps of the 2nd floor of the MIT Stata Center were built. After that, images of the probabilistic representations of these maps were obtained. On these images, about 1000 features were detected using of various detectors. Next, the descriptors were computed. The imprecise count of the detecting keypoints is due to the fact that in OpenCV the BRISK and SURF algorithms do not provide the ability to explicitly specify the number of points. This count can only be influenced through some of the input parameters.

The table I shows the ratios of correctly matched feature points to the total number of matched feature points. Each line corresponds to a pair of different maps to merge. Repeated running with averaging of the results were not carried out, because algorithms for detecting and calculating descriptors do not depend on random variables.

The columns with the names of the algorithms indicate the coefficients that show the ratio of correctly matched pairs to the total number of matched pairs of features. As you can see from the table, the ratio of points that the ORB detects is better, or close to the best, so this feature detector was chosen.

D. Maps alignment

After calculating the descriptors, the found points must be matched. For this, brute force matching is used using the Hamming norm [18]. This method is the main one for matching binary keypoints descriptors. The Lowe test [17] is then used to partially eliminate false matches. After filtering, the matched pairs that passed the test are copied into two vectors of features that correspond to their map. The optimal affine transformation is calculated between these vectors. The result of the function operation is the 2×3 matrix $[R|t]$, where R is the rotation matrix 2×2, t is the translation vector 2×1.

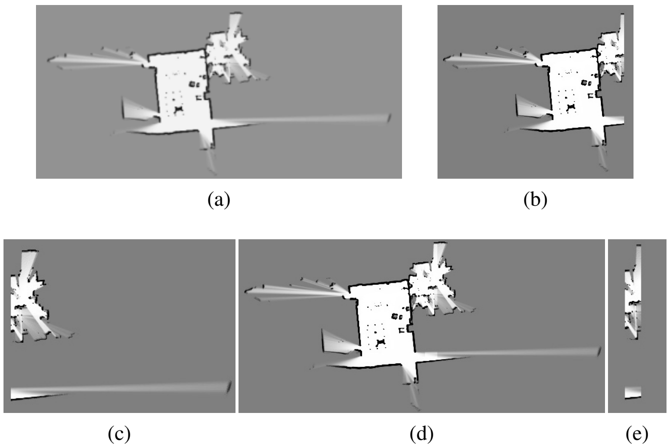


Figure 2. Comparison of conjunctive and disjunctive rules. (a) - the original map, (b) - 51 % of the left side of the map, (c) - 51 % of the right side of the map, (d) - the map combined using the conjunctive rule, (e) - the map combined using disjunctive rule

The found transformation is applied to the map for which it was calculated.

E. Merging using the disjunction rule

The conjunctive rule is used if all sources are considered reliable, disjunctive – if at least one source is considered unreliable. Maps, which are built by robots, can have various anomalies, such as incorrect scales (a robot in some area covered more/less distance than shown on the map), inconsistency of angles on the map and in reality (rotation error), and others. Therefore, it was decided to use the disjunctive rule, which allows you to exclude areas that have strong differences from the resulting map. For a comparison of conjunctive and disjunctive rules, see V-A.

Conjunctive rule:

$$m_{1 \cap 2}(A) = \sum_{B \cap C = A} m_1(B)\, m_2(C) \tag{5}$$

Disjunctive rule:

$$m_{1 \cup 2}(A) = \sum_{B \cup C = A} m_1(B)\, m_2(C), \tag{6}$$

where $A, B, C \in 2^X \neq \emptyset$.

After the maps are aligned, they are merged using the disjunctive rule (6). Since the maps are aligned, an iterative traversal of the cells occurs. For each pair, the specified rule is applied, and the information obtained during the merging is set in a cell with the same index as those taken in the original maps.

V. Experiments

The algorithm for searching for transformation and merging of maps was described above. It has been said that there are different merge rules in TBM. As part of the work, conjunctive and disjunctive rules are considered. The reasons for choosing a disjunctive merge rule will be presented below.

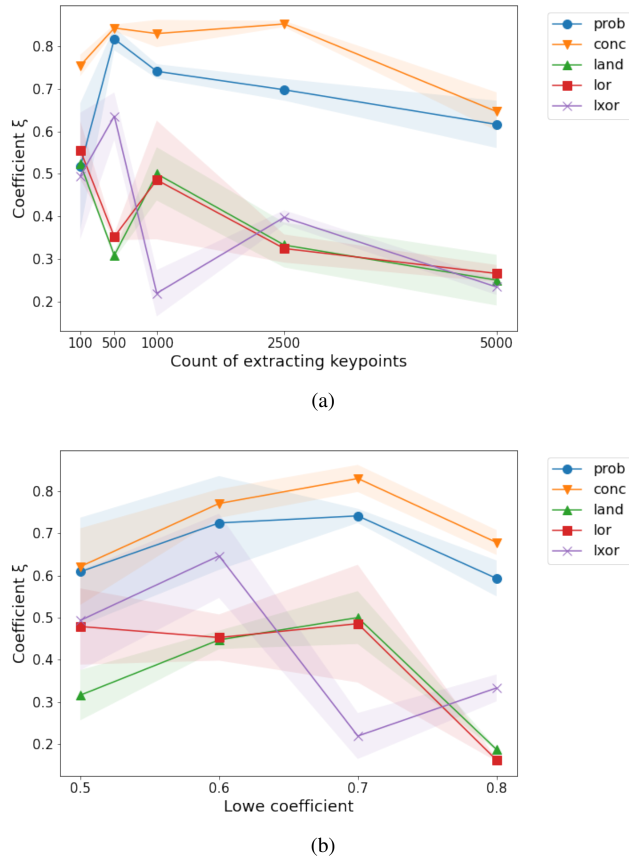


Figure 3. Graph of the ratio of correctly matched keypoints to the total number of matched keypoints depending on (a) - the initial number of extracting keypoints(the value of the Lowe coefficient - 0.7), (b) - the value of the Lowe coefficient (the number of extracting keypoints - 1000)

In the section IV-B it was said that descriptor concatenation is used. This section will look at the various modifications to existing descriptors and compare these modifications. The accuracy of the found transformation is determined by finding the distance between the coordinates of the keypoints of the first map and the coordinates of the keypoints of the second map after applying the found transformation. This will be discussed below.

A. Comparison of conjunctive and disjunctive rules

As mentioned above, the conjunctive rule is used when all sources are considered reliable, and the disjunctive one is used when at least one source is not. The difference between merges using conjunctive and disjunctive rules is considered in the following example. The original map is shown in fig. 2a. Fig. 2b - 2c are represented by 51% of the left and right parts of the original map with the same size of the original map, the remaining 49 % are set as cells, about which there is no information. Fig. 2d is demonstrated the result of merging using the conjunction rule, so the resulting map contains the parts that are present on both merged maps. The cells that are in the overlapping area have reduced the mass of the belief corresponding to the lack of information about this cell and redistributed it according to the rule (5). This can be seen

Table II
COMPARISON OF DIFFERENT DESCRIPTOR MODIFICATIONS

1	2	3a	3b	3c	3d	3e	4
100	0.5	0.53 ± 0.10	0.55 ± 0.08	0.52 ± 0.15	0.34 ± 0.09	0.32 ± 0.10	a,b,c
	0.6	0.53 ± 0.11	0.74 ± 0.04	0.67 ± 0.02	0.52 ± 0.08	0.38 ± 0.10	b
	0.7	0.52 ± 0.15	0.75 ± 0.03	0.52 ± 0.01	0.56 ± 0.07	0.50 ± 0.15	b
	0.8	0.62 ± 0.02	0.78 ± 0.02	0.35 ± 0.02	0.55 ± 0.02	0.58 ± 0.09	b
500	0.5	0.75 ± 0.09	0.76 ± 0.04	0.49 ± 0.08	0.63 ± 0.11	0.42 ± 0.13	a, b
	0.6	0.70 ± 0.11	0.77 ± 0.04	0.43 ± 0.03	0.56 ± 0.05	0.56 ± 0.10	a, b
	0.7	0.82 ± 0.03	0.84 ± 0.01	0.31 ± 0.01	0.35 ± 0.01	0.64 ± 0.06	a, b
	0.8	0.66 ± 0.03	0.65 ± 0.04	0.18 ± 0.01	0.23 ± 0.01	0.39 ± 0.02	a, b
1000	0.5	0.61 ± 0.13	0.62 ± 0.09	0.32 ± 0.06	0.48 ± 0.09	0.49 ± 0.12	a, b
	0.6	0.72 ± 0.11	0.77 ± 0.03	0.45 ± 0.02	0.45 ± 0.05	0.65 ± 0.10	a, b
	0.7	0.74 ± 0.02	0.83 ± 0.03	0.50 ± 0.06	0.49 ± 0.14	0.22 ± 0.05	b
	0.8	0.59 ± 0.04	0.68 ± 0.03	0.19 ± 0.00	0.16 ± 0.00	0.33 ± 0.03	b
2500	0.5	0.69 ± 0.11	0.63 ± 0.11	0.28 ± 0.03	0.57 ± 0.08	0.26 ± 0.10	a
	0.6	0.76 ± 0.07	0.78 ± 0.02	0.30 ± 0.01	0.40 ± 0.04	0.62 ± 0.02	a, b
	0.7	0.70 ± 0.03	0.85 ± 0.01	0.33 ± 0.05	0.32 ± 0.03	0.40 ± 0.02	b
	0.8	0.54 ± 0.06	0.58 ± 0.05	0.30 ± 0.10	0.16 ± 0.01	0.24 ± 0.02	a, b
5000	0.5	0.69 ± 0.09	0.90 ± 0.03	0.18 ± 0.03	0.31 ± 0.06	0.49 ± 0.09	b
	0.6	0.83 ± 0.01	0.83 ± 0.03	0.21 ± 0.03	0.26 ± 0.03	0.48 ± 0.03	a, b
	0.7	0.62 ± 0.06	0.65 ± 0.05	0.25 ± 0.06	0.27 ± 0.02	0.23 ± 0.02	a, b
	0.8	0.49 ± 0.07	0.51 ± 0.06	0.26 ± 0.02	0.17 ± 0.01	0.23 ± 0.01	a, b

if attention is paid to the overlapping 2% vertically of the resulting map – the color has become whiter, that is, the mass of the belief that the cell is free, and therefore the likelihood that this cell is free, has increased. Fig. 2e is shown the result of merging using the disjunction rule. So only that part of the original map that is present on both merged maps gets into the resulting map. The choice of the disjunctive rule is explained by the fact that the absence of information is of higher priority than the presence of unreliable information.

B. Comparison of descriptor modifications

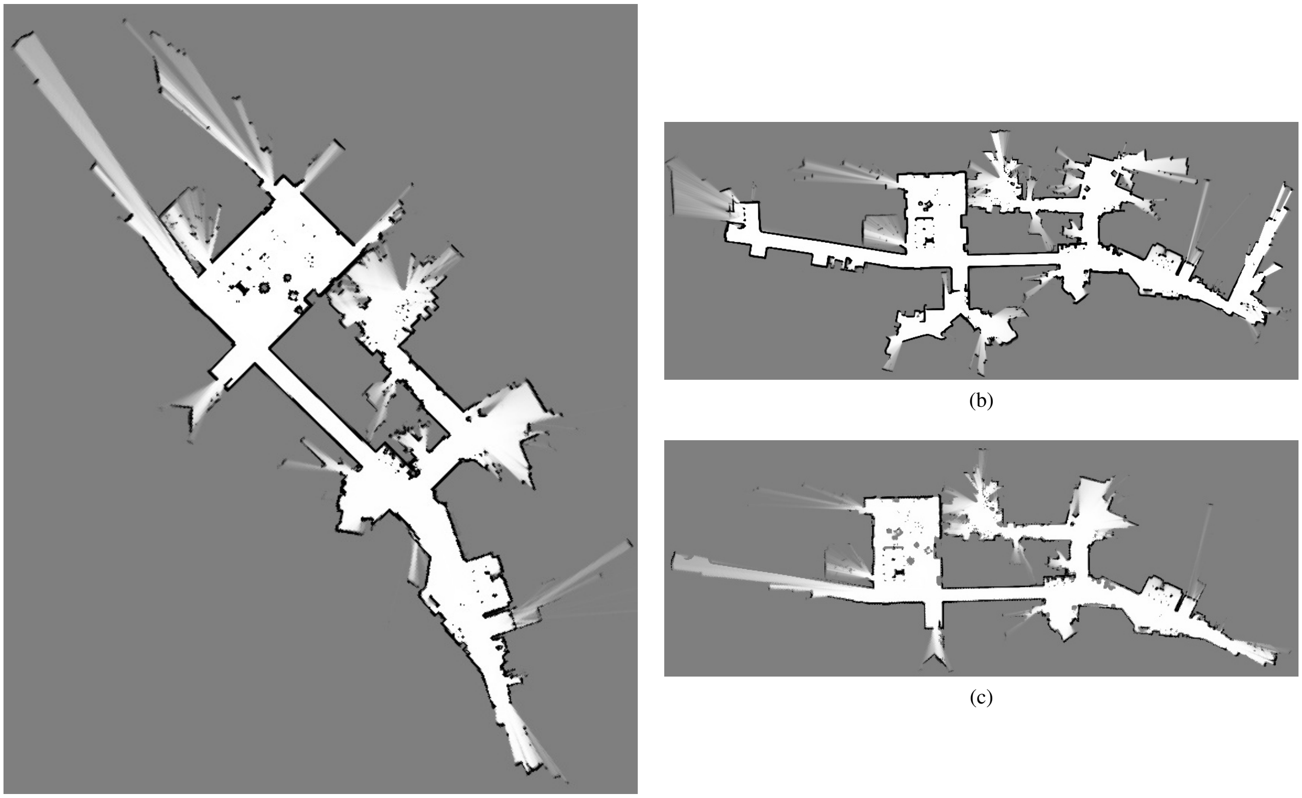
Comparative analysis of different modification of the 128-bit ORB descriptor can be seen in the table II. The columns contain the following information: 1 – the number of keypoints to be detected initially; 2 – Lowe coefficient used when filtering false matches; 3 – the ratio of the number of correctly matched features to the total number of matched features ξ (experiments were carried out on 10 different pairs of maps); 4 – the columns showing the highest coefficients indicated in columns 3. Modifications in column 3: a – approach without modification of descriptors; b – descriptors concatenation: for each keypoint, three 128-bit descriptors are concatenated, resulting in a 384-bit descriptor; c – using the logical and operation: applying a bitwise logical and for three 128-bit descriptors, the result is a 128-bit descriptor; d – applying of a logical or operation: applying of a bitwise logical or for three 128-bit descriptors, the result is a 128-bit descriptor; e – applying a logical exclusive or operation: applying a bitwise logical exclusive or for three 128-bit descriptors, resulting in a 128-bit descriptor.

It can be concluded that the concatenation of feature point descriptors obtained using TBM almost always shows a greater value of the ξ ratio than any other proposed modification. In most cases the value of the ratio ξ when using concatenation is not less than when matching the descriptors of keypoints calculated for the probabilistic representation of occupancy maps.

The dependence of the ratio ξ on the number of initially extracted features (at a fixed value of the Lowe coefficient) can be seen in fig. 3a. Based on this graph, statistically significant differences were seen with a confidence level of 0.95 when extracting 100, 1000 and 2500 keypoints. In the implementation of the algorithm, the default value is set to 1000. The dependence of the ratio ξ on the value of the Lowe coefficient (with a fixed number of initially extracted keypoints) can be seen in fig. 3b. Based on this graph, statistically significant differences with a confidence level of 0.95 were seen with a coefficient value of 0.7 or 0.8. In the implementation of the algorithm, the default value is 0.7.

C. Accuracy of the transformation

Viny_SLAM [19] allows building occupancy grid maps based on TBM. Compared to analogs such as FastSLAM [20] or Credibilist SLAM [9], Viny_SLAM is verified on the MIT Stata Center dataset. Using the specified SLAM algorithm, various maps of the 2nd floor of the MIT Stata Center were obtained. After that, the algorithm proposed in this work was applied to pairs of maps. During the operation of the algorithm, the keypoints of the maps were extracted and matched. Then the calculated transformation was applied to one of the sets of keypoints. For pairs of matched points, the distances between



(a)

(b)

(c)

Figure 4. The result of the algorithm. (a) - first merge map, (b) - second merge map, (c) - result of merge

Table III
DISTANCES BETWEEN MATCHED KEYPOINTS OF MAPS AFTER TRANSFORMATION

Maps pair	min, m	max, m	gmean, m
1	0.031	16.043	0.158
2	0.000	0.000	0.000
3	0.028	0.707	0.138
4	0.012	18.058	0.330
5	0.029	30.904	0.256
6	0.017	0.854	0.098
7	0.006	26.217	0.121
8	0.011	22.701	0.104
9	0.004	33.530	0.305

Table IV
ALGORITHM RUNNING TIME

Maps pair	area, m^2	mean, s	std, s
1	8700	971.5	98.142
2	12100	1395.9	117.230
3	13200	1501.4	146.242
4	4125	446.4	24.707
5	4900	598.5	48.028
6	5625	692.3	70.477
7	12100	1420.9	96.175
8	12075	1370.3	101.308
9	12100	1365.5	115.612

these points were calculated. The table III shows the minimum, maximum and geometric mean distances in meters between the matched feature points after applying the transformation. Repeated experiments gave the same values. Based on the values of the geometric mean in the table, it can be concluded that the matching error does not exceed 0.35 meters. This error is due to the fact that not all keypoints are matched correctly.

Fig. 4 shows the maps to be merged and the result of the algorithm. Maps are shown in probabilistic representation. Map cell size is $0.1 \times 0.1 m^2$. One pixel on the image of the map used to construct descriptors corresponds to one cell, thus one pixel corresponds to $0.01 m^2$.

Table IV lists the areas of the maps resulting from this experiment, as well as the average fusion time and standard deviation for 10 experiments for each pair of maps. The processing time was calculated on an Intel Core i5-8300H 2.3 GHz processor. This operating time is much longer than that required for real-time operation, but the algorithm was not originally intended for such use. Within the framework of the considered analogs, the proposed algorithm shows a faster operating time. One of the possible applications can be the construction of models of buildings in which people are dangerous. This can be used to evaluate the structure for subsequent deconstruction.

VI. Conclusion

In this paper, a TBM cell representation of an occupancy grid map has been described. This cell allows you to store more information about the space than a probabilistic representation. It also was compared different detectors of keypoints on the map image in order to choose the one that would be used in the algorithm. As a result, ORB was chosen. The reason is that the ratio of correctly matched pairs to the total number of matched pairs of features was higher or close to the highest. After that, various possible modifications of descriptors were considered to improve the accuracy of matching of keypoints. As a result, it was concluded that concatenation gives better results than any other modification, and also in most cases no worse than the original descriptor. Both disjunctive and conjunctive rules in TBM could be used to merge maps. However, preference was given to the disjunctive rule, since the maps may contain various errors, and this rule allows you to cut off inconsistent areas. At the end, the results of the algorithm were presented, as well as an assessment of the accuracy of the transformation search.

As a further development of this work, the clustering of keypoints and the verification of the existence of a transformation between clusters of two maps is considered. If a transformation is found, then it is possible to apply the transformation to an area of the map containing a cluster of special points. After applying a transformation to a part of the map, it is possible to merge using one of the merge rules in the Transferable Belief Model.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[3] K. Sentz, S. Ferson *et al.*, *Combination of evidence in Dempster-Shafer theory*. Sandia National Laboratories Albuquerque, 2002, vol. 4015.

[4] P. Smets and R. Kennes, "The transferable belief model," *Artificial intelligence*, vol. 66, no. 2, pp. 191–234, 1994.

[5] G. Shafer, *A mathematical theory of evidence*. Princeton university press, 1976, vol. 42.

[6] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1384–1397, 2006.

[7] H. Li, M. Tsukada, F. Nashashibi, and M. Parent, "Multivehicle cooperative local mapping: A methodology based on occupancy grid map merging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2089–2100, 2014.

[8] S. Saeedi, L. Paull, M. Trentini, M. Seto, and H. Li, "Group mapping: A topological approach to map merging for multiple robots," *IEEE Robotics & Automation Magazine*, vol. 21, no. 2, pp. 60–72, 2014.

[9] G. Trehard, Z. Alsayed, E. Pollard, B. Bradai, and F. Nashashibi, "Credibilist simultaneous localization and mapping with a lidar," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2699–2706.

[10] P. Smets, "Data fusion in the transferable belief model," in *Proceedings of the third international conference on information fusion*, vol. 1. IEEE, 2000, pp. PS21–PS33.

[11] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.

[12] K. P. Murphy *et al.*, "Bayesian map learning in dynamic environments." in *NIPS*, 1999, pp. 1015–1021.

[13] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 1. IEEE, 2003, pp. 206–211.

[14] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[15] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.

[16] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] B. Waggener, W. N. Waggener, and W. M. Waggener, *Pulse code modulation techniques*. Springer Science & Business Media, 1995.

[19] A. Huletski, D. Kartashov, and K. Krinkin, "Vinyslam: an indoor slam method for low-cost platforms based on the transferable belief model," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6770–6776.

[20] T. Reineking and J. Clemens, "Evidential fastslam for grid mapping," in *Proceedings of the 16th International Conference on Information Fusion*. IEEE, 2013, pp. 789–796.