Building of 3D Environment Models for Mobile Robotics Using Self-Organization

Jan Koutník[1], Roman Mázl[2] and Miroslav Kulich[2]

[1] Neural Computing Group, Department of Computer Science and Engineering,
Faculty of Electrical Engineering,
Czech Technical University in Prague, Karlovo nam. 13, 121 35 Prague 2, Czech Republic
http://ncg.felk.cvut.cz
koutnij@fel.cvut.cz
[2] Center of Applied Cybernetics,
Faculty of Electrical Engineering,
Czech Technical University in Prague, Technická 1, 166 27 Prague 6, Czech Republic
{mazl, kulich}@labe.felk.cvut.cz

Abstract. In this paper, we present a new parallel self-organizing technique for three dimensional shape reconstruction for mobile robotics. The method is based on adaptive input data decomposition, parallel shape reconstruction in decomposed clusters using Kohonen Self-Organizing Map, which creates mesh representation of the input data. Afterwards, the sub-maps are joined together and the final mesh is re-optimized. Our method overcomes a problem of fitting one mesh to complex non-continuous shapes like building interiors. The method allows to process unordered data collected by mobile robots. The method is easily paralelizable and gives promising results.

1 Introduction

The problem of surface reconstruction appears in many applications of mobile robotics as well as in geometric model acquisition in computer graphics and computer vision. Our objective is to build environment models from the real measured data of building interiors. The data has been collected using mobile robot equipped with laser range-finder. The data set is a set of unorganized points.

There exist a lot of methods for the 3D reconstruction, see e.g. [1, 2]. The problem of the reconstruction can be expressed as a procedure of learning the topology from the data set and reduction of the data set cardinality. Artificial neural networks are commonly used for solution of this task. We already implemented and tested Neural Gas algorithm for reconstruction of two dimensional environmental map [3]. Nice application of the Kohonen Self-Organizing Map (SOM) [4] can be found in [5], where a top-down approach (improving already known topology of the data using SOM networks) in reconstruction instead of bottom-up (building connection among nearest points) is used. The Kohonen

SOM is a suitable approach to learning the topology. The SOM network itself is usually a two-dimensional mesh, which self-organizes to cover the data during the learning phase. The surface data are two-dimensional, thus a usage of SOM is a good approach. Another approach based on Kohonen SOM can be found in recent work [6].

The main problem of the presented solutions is that a single SOM is normally used for the reconstruction. The topology of the SOM is normally a rectangular mesh, which stretches during the learning phase. In most applications usage of a rectangular mesh is a correct approach but usage of such mesh for representation of a complex surface in three dimensions does not give precise results.

Imagine that you let the SOM network to reconstruct the human body surface. The resulting shape will be a strange clothing which will not perfectly fit the body and it will not be easy to wear it. Our approach is similar to a work of a tailor which has to sew a perfect fit dress from a stretch fabric. The input space (a human body) is divided to several parts (legs, trunk, arms etc.) and the tailor cuts the correct shapes from the two-dimensional fabric. The parts are sewn together and the dress is ready. The stretch material allows some deformations to fit the body when used. Our algorithm works similarly.

This paper is organized as follows. Section 2 describes how the data sets were obtained. Section 3 describes proposed reconstruction algorithm. Section 4 shows experimental results. Section 5 contains a discussion of the results. Section 6 concludes the work and section 7 mentions several improvements that will be made in near future.

2 Data Acquisition

All experimental data has been gathered using the G2 mobile robot developed in the Gerstner laboratory, see figure 1. The G2 robot is non-holonomic differential drive robot for indoor usage able to reach the maximal velocity about 0.5 m/s. The robot motion is established by two stepper motors.

2.1 Orthogonally mounted range-finders

The former approach of data gathering uses the G2 mobile robot with two orthogonally mounted SICK LMS 200 laser range-finders as can be seen in figure 1.

The first laser range-finger has been mounted horizontally, while the second laser range-finder has been mounted in vertical position pointing up perpendicularly to the horizontal measurement half-plane of the first laser. The horizontal laser measures data in parallel half-plane to the floor in order to ensure proper 2D localization of the robot in space. The data obtained from the horizontal laser as well as the robot trajectory during acquisition of experimental data are shown in figure 2.

The knowledge of the current position of the mobile robot allows registration of the spatial data from the vertical laser. The measured data from the vertical

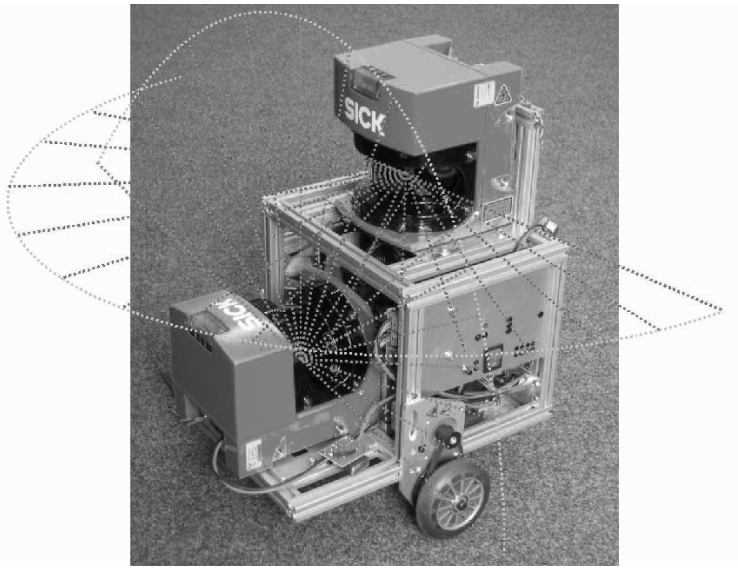

Fig. 1. Hardware setup of the G2 mobile robot

laser describes depth of surrounding environment while the robot moves. This configuration in fact replaces expensive 3D scanners for spatial point retrieval.

2.2 Data registration

Knowledge of the correct position of the robot (the place from where the data has been measured) is the essential condition for future data processing. The localization issues are one of the central topic in mobile robotics, an overview can be found e.g. in [7] or [8]

Our solution of the localization problem is based on modified scan matching technique utilizing the ICP (Iterative Closest Point) algorithm [9–11]. The modification is targeted on improvements of outlier point rejection and on combination of the scan matching technique with a global point based map of the environment. The usage of the simultaneously built map increases precision and reliability of the scan matching technique during a localization process.

Two different approaches to the registration of 3D data has been used, since we have two sensor setup of the robot. The first setup employs a 2D version of ICP localization algorithm that processes only laser scans incoming from the horizontal range-finder while the robot is continuously traversed through the environment and gathers data from the vertical laser range-finder. It means that registration of the vertical spatial scan is just determined by the proper localization of the whole mobile robot using horizontal range-finder. The precision of localization is approx. 5cm.

The second sensor setup with the sweeping laser assumes the stop-and-go style of mobile robot movements since a complete 3D scan has to be measured

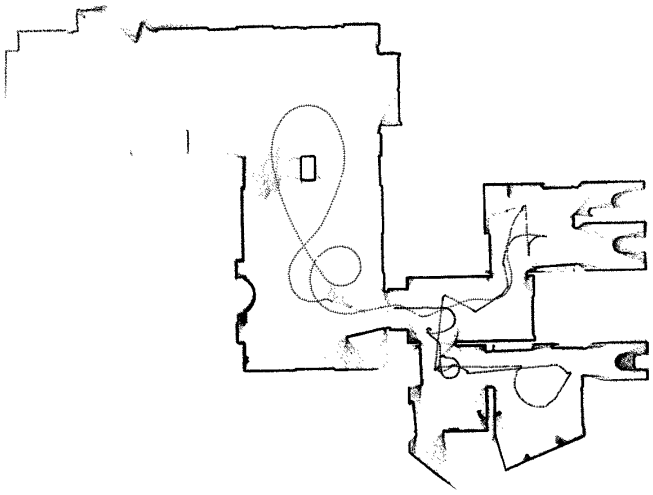

Fig. 2. Data collected from the horizontal laser and trajectory of the robot.

from one position. The current raw 3D scan position is estimated using a robot odometry. The final fine alignment of the 3D scans in 6DOF ensures the 3D version of ICP algorithm, where a kD-tree approach was used for searching for corresponding pair points. The calculation of a transformation for 3D rotation and translation minimizing the mutual scan misalignments is realized by a quaternion approach. All computation details can be found in [12]. With a knowledge of the sensor position a set of the 3D laser scan can be merged together in order to obtain a huge 3D point cloud of spatial points that describes a shape of a surrounding environment.

3 Algorithm for 3D Reconstruction

We designed a hybrid algorithm that embeds either non-topological clustering techniques like Neural Gas and K-Means algorithms for initial data preprocessing and topological clustering algorithm – Kohonen SOM for building of wired model of the data. The main idea behind is that several SOMs are executed in parallel and process clustered input data. We introduce a joining phase, which combines the SOM output together into large SOM, which is partly re-optimized. Thus, the algorithm has following four phases:

1. *initial clustering,*
2. *building of sub-maps,*
3. *joining phase,*
4. *re-optimization.*

3.1 Initial Clustering

The basic idea is to divide the input data into subsets to ease usage of SOM algorithm for the subsets. We used algorithms, which preserve the distribution of the data in the input space. This approach has two advantages. First, as mentioned in introductory part of this paper, the data needs to be clustered to allow proper approximation of the data by two-dimensional mesh. Second, such clustering rapidly reduces the real time complexity of the method. It allows the second phase to run in parallel. The complexity of SOM algorithm rapidly grows with the map size due adaptation of neighboring neurons in the map. Sequential execution of SOMs with the same sum of neurons is even faster than execution of one big SOM.

The clustering has another property exploited by the method. The SOMs in the second phase are put only in the parts of the input space where the data exists. If one starts to approximate the whole input space with one big SOM, the SOM neurons will appear in the space where no data are present. Such approach does not allow to process two isolated objects in the scene like boxes and obstacles, which appear in the robot workspace. Our approach eliminates this drawback.

We used Neural Gas (NG) [13] and K-Means [14] algorithms. The unsupervised clustering algorithm splits data into clusters. Afterwards, in the second sub-map building phase, the individual SOMs are used for topological approximation of the data in the clusters separately.

First, we employed the Neural Gas (NG) algorithm for data clustering. The complexity of the NG is $O(n*d*m*k*log(k))$ where n is the cardinality of the data set, d is a dimension of the input data (3), m is number of iterations over the data set and k is required number of clusters (number of neurons in NG). The NG algorithm requires the neurons to be sorted after each presentation of the input vector, which is expressed by $k*log(k)$.

The most crucial element is the number of iterations. NG algorithm requires about 40000 iterations to get proper clustering. Our data sets have typically more than 100 thousands of data vector, which makes the NG algorithm unusable due it's big time complexity.

The K-Means (KM) algorithm has complexity $O(n*d*m*k)$ but it requires to be m much less than in the case of the NG. Practically, the NG algorithm takes several hours but the KM ends in few minutes (measured on 2 GHz Opteron processor).

There is virtually no difference between clusters obtained with NG and KM algorithm. Thus we use computationally optimal K-Means in following experiments.

3.2 Building of Sub-maps

After the initial clusters are created, the SOM is built from data subsets. We use well known iterative Kohonen SOM algorithm described in [4].

The neurons in the SOM are organized in either rectangular or hexagonal map. We use the rectangular mesh. Two distances are defined within the algorithm. The first one, the Euclidean distance, is measured between weights of the neuron (vector stored in the neuron) and an input vector. The second distance is measured in the SOM map. Closest neighbors have the distance defined to be one, distance of other neurons is defined by the shortest path between the neurons in the map. In the learning phase, so called Best Matching Unit (BMU) is selected among all neurons. It is a neuron with closest Euclidean distance to the input vector. The neuron weights are updated to be closer to the input vector. Weights of the neurons in the neighborhood are updated as well. Closest neurons are updated more than far neurons. The amount of learning is expressed by a function of distance in the map.

This property makes the SOM to be suitable for our approach. The SOM algorithm places a mesh in three dimensional data in order to bet best coverage of the input data. In contradiction to first phase of the algorithm, the clustering, we need the network to be topological.

The advantage is that the sub-maps can be created in parallel. The clustering algorithms in the first phase creates clusters with approximately similar cardinality. Thus the SOM algorithms in the second phase take relatively same computational time. We do not have to solve other parallelization issues like load balancing etc. Only the initial distribution of the load is needed.

3.3 Joining Phase

In the joining phase, all the SOMs created in the previous phase are merged together. We use a simple nearest neighbor algorithm where for each neuron in the SOM boundary a candidate in other SOM boundaries is searched. If the candidate has an Euclidean distance lower than a specified threshold, the connection between the two neurons is built. The threshold is a mean distance to closest neighbors of the candidates in their SOM maps ($\bar{d}$) multiplied by a constant (θ) as depicted in figure 3.

The connections obtained in the joining phase are exploited in the last re-optimization phase. The joining phase creates large SOM from the particular SOMs. The added connections are taken as normal connections in the standard SOM algorithm. The resulting SOM has not a planar topology.

3.4 Re-optimization

It the last phase the final SOM is re-optimized. The key point is that the final SOM is not a planar structure. It has a desired structure corresponding to the data distribution in the input space. Our implementation of the SOM allows us to work with such SOM maps which are not strictly rectangular. The neighborhood neurons are stored in list of neighbors. The joining phase processes these lists and adds neurons from the joined SOMs.

The SOM algorithm used in this phase contains a little modification. It is not necessary to re-optimize the whole map. If so, the problems with the time

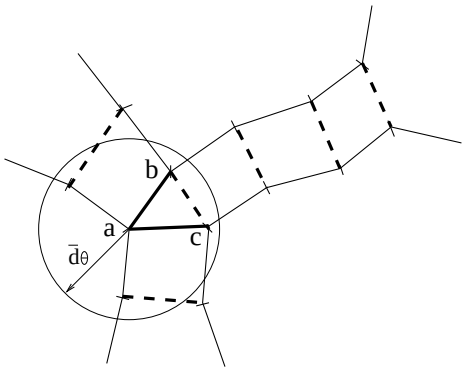

Fig. 3. Joining phase. The map boundaries alre depicted with solid line. Two connections are built for neuron a to neurons b and c, whose distance to neuron a is lower than the threshold ($\bar{d}\theta$). Dashed lines express connections between other neurons constructed by the same algorithm.

complexity will arise again. Thus we re-optimize only that neurons, which are close to boundaries of the joined sub-maps. The neurons that will not be re-optimized as well as the corresponding data are removed from the calculation before the re-optimization starts. This speeds up the algorithm.

The final SOM algorithm optimizes joins of the maps. It precises the joining, makes approximation of edges to be sharper and better fit the input data.

4 Experimental Results

As the experimental data set we used a set of points measured in the first floor of our building. Figure 2 shows the trajectory of the robot and the data collected from the horizontal laser in purpose of better understanding to reconstructed three-dimensional model shown in figure 4. All phases of the algorithm were performed on the data. The data contained approximately 126 thousands of points, in the first phase 100 clusters were created, the sub-maps contained 10x5 neurons. The algorithm run takes about 2 hours on the 2 GHz Opteron processor for the complete data set when executed sequentially. Physical parallelization of the second phase would save at least one hour of computational time. Other parameters were setup the same in all experiments. All SOM maps iterated 10000 times over all input data, the initial learning rate was setup to 0.5 and diminished to 0.005 linearly during the learning. The neighborhood was initially setup to the longest path in the map and diminished to 1. The joining parameter θ was experimentally setup to 1.8.

The algorithm works also if the data do not consist of continuous shapes. We created an artificial data set with two artificially generated cubes, each consisting of 726 points. We used 12 clusters and SOM sub-maps of 10x5 neurons. We can

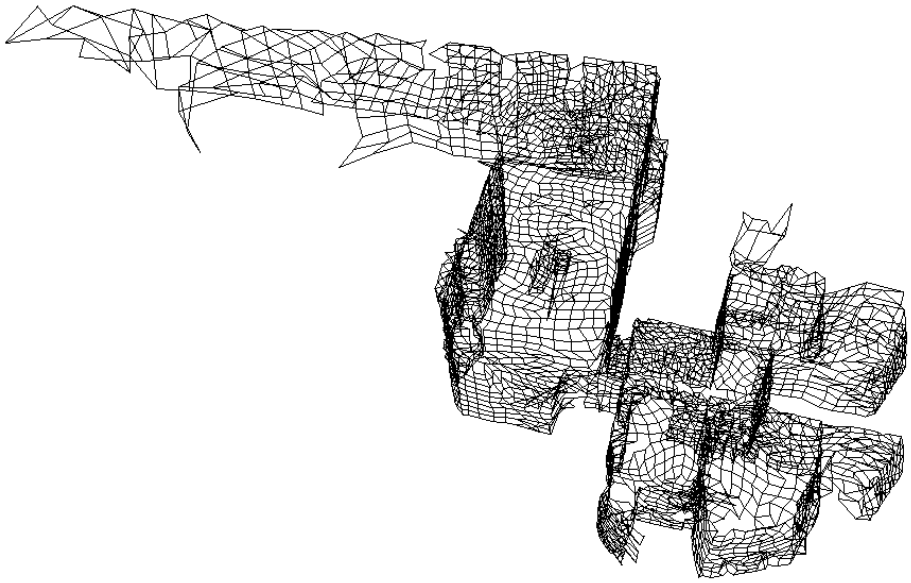

Fig. 4. Final algorithm result – the reconstructed three-dimensional interior from the vertical laser data. The rooms with a lower ceiling on the right-hand side are toilets. The L-shape corridor is on the left-hand side. The front wall of the corridor contain low density of input points as can be seen also in the horizontally scanned data.

see that each cube was easily encapsulated into 6 sub-maps joined together and re-optimized. The result is shown in figure 5.

Figure 6 shows the reconstruction using a single SOM consisting of same amount of neurons as in previous example. The SOM had 20x30 neurons. Single SOM could not split itself into two parts. Usage of one SOM for complex data reconstruction could not be satisfactory. The computational complexity of such large mesh caused the algorithm to run approximately 30 minutes in comparison to 6 minutes of run of our algorithm including re-optimization of the final map.

5 Discussion

The clustering and decomposition of the input space to clusters using K-Means and reconstruction in clusters using Self-Organizing Map seems to be promising combination of algorithms for the problem of three-dimensional shape reconstruction. As can be seen in figures 4 and 5, there are still some holes in the joined maps. This is caused because the optimal join of the neighboring sub-maps is not straightforward. The grids of the maps can be mutually rotated and a search for the correct join is also an optimization problem. Our method provides a wired model. If we want to obtain a solid model a triangulation has to be performed. A triangulation of the sub-maps is easy, each rectangle is split

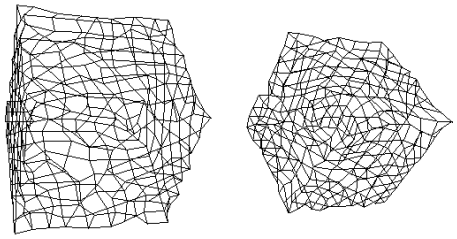

Fig. 5. Reconstruction from the data sampled to two isolated cube shapes. Our algorithm arranged 6 sub-maps to each cube.

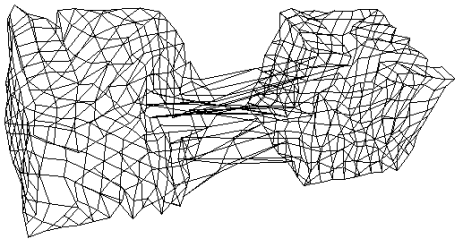

Fig. 6. Reconstruction from the data sampled to two isolated cube shapes using a single SOM map. It can be seen, that the cubes were not isolated. Even if the neurons are placed to the data clouds the connections between map still exist and disrupt the final reconstruction. Reconstruction of at least two objects in the scene fails using a single connected structure.

into two triangles. Triangulation of the joins is not that obvious. Currently, the joins are exploited as a *logical* connections between SOMs. Such joins allows the final map to be more complex than a planar rectangular mesh only. The triangulation of stripes between the maps requires embedding of another algorithm for triangulation of stripes.

We currently use a nearest neighbor algorithm to produce the joins between sub-maps. The algorithm is simple and produces joins that are used in the final re-optimization phase. The threshold for joining the neighbors on the map borders was experimentally found to be 1.8 times an average Euclidean distance between joined neurons and their neighbors. If the constant is lower than 1.8 the sub-maps are not enough joined. If the constant is higher, the map contains crossed joins, the final SOM algorithm deforms the joined parts of the sub-maps and the further triangulation would not be easy.

The final re-optimization precises the final mesh. The time complexity of the final phase is reduced by a selection of the input data and neurons that are on the sub-maps' boundaries.

6 Conclusion

In this paper we introduced a combination of self-organizing techniques that gives satisfactory results in the task of reconstruction of shapes from the unordered data sets. The algorithm was tested on data obtained from a laser range-finder sensor mounted on a mobile robot. It was shown that a usage of only one algorithm, either Neural Gas or Self-Organizing Map, does not lead to good results for reconstruction of complex and non-continuous shapes. The idea of the initial decomposition, building of separate sub-maps using topological Kohonen Self-Organizing Map, joining and re-optimization gives nice and stable results. The method reduces the complexity of the data and produces a wired model of the environment where the robot is situated. The algorithm allows to create model of more than one isolated objects found in the input data. The second phase of the algorithm can be effectively parallelized. The reconstruction can be built using collaboratively scanning robots, since the algorithm allows processing of unorganized data sets.

7 Future Work

Future research will focus on improvements of the current algorithm. The K-Means clustering algorithm will be replaced with an algorithm which preserves the flat clusters so the clusters will represent the walls found in the data sets. A usage of triangular shape SOM map is expected instead of the rectangular topology. The batch version of SOM training algorithm can rapidly diminish the computational time as well. The map itself will create the triangulation needed for solid displaying of the reconstructed shape. Finally, an optimizing algorithm for generation of joins will be embedded instead of simple and computationally inexpensive nearest neighbor joining algorithm.

Acknowledgment

This work has been supported by the Ministry of Education of the Czech Republic under Projects No. 1M0567 and MSM6840770012. The implementation of the algorithms was performed by CTU student Tomáš Jaroš.

References

1. Sequeira, V., Ng, K.C., Wolfart, E., Gonçalves, J., Hogg, D.: Automated 3d reconstruction of interiors with multiple scan-views. In: Proceedings of SPIE, Electronic Imaging '99, IS & T/SPIE's 11th Annual Symposium, San Jose, CA, USA (1999)
2. Christian Früh, A.Z.: An automated method for large-scale, ground-based city model acquisition. International Journal of Computer Vision **60**(1) (2004) 5–24
3. Štepán, P., Kulich, M.: Buildings maps for autonomous mobile robots. In: Field and Service Robotics. Helsinki: FSA-Finnish Society of Automation. Volume 1. (2001) 321–326

4. Kohonen, T.: Self-Organizing Maps. 3rd edn. Springer-Verlag (2001)
5. Yu, Y.: Surface reconstruction from unorganized points using self-organizing neural networks (1999)
6. Farid Boudjema, Philippe Biela Enberg, J.G.P.: Dynamic adaptation and subdivision in 3d-som: Application to surface reconstruction. In: 7th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05). (2005) 425–430
7. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. The MIT Press (2005)
8. Siegwart, R., Nourbakhsh, I.R.: Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents). The MIT Press (2004)
9. Lu, F., Milios, E.: Robot pose estimation in unknown environments by matching 2D range scans. In: CVPR94. (1994) 935–938
10. Besl, P., McKay, N.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **2**(14) (1992) 239–256
11. Mázl, R., Přeučil, L.: Methods for mobile robot localization. In: 5th International Scientific - Technical Conference Process Control 2002, ŘÍP 2002, University of Pardubice (2002)
12. Surmann, H., Nchter, A., Hertzberg, J.: An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. Journal Robotics and Autonomous Systems **45**(3-4) (2003) 181– 198
13. Fritzke, B.: Growing cell structures—a self-organizing network in k dimensions. In Aleksander, I., Taylor, J., eds.: Artificial Neural Networks, 2. Volume II., Amsterdam, Netherlands, North-Holland (1992) 1051–1056
14. Christopher, B.: Neural networks: A pattern recognition perspective (1996)