THESIS ON MECHANICAL ENGINEERING E76

Global 3D Map Merging Methods for Robot Navigation

DMITRY SHVARTS

TUT PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Mechanical Engineering
Department of Mechatronics

Dissertation was accepted for the defense of degree of Doctor of Philosophy in Engineering on 21 may, 2013.

Supervisor: Prof. Mart Tamre
Department of Mechatronics, Tallinn University of Technology

Opponents: Prof. Victor M. Musalimov
St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Russia

Associate Professor Javier Civera
University of Zaragoza, Spain

Defense of the thesis: 27 june, 2013

Declaration:
Herby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for any other degree.

/Dmitry Shvarts/

Copyright: Dmitry Shvarts, 2013
ISSN 1406-4758
ISBN 978-9949-23-493-6 (publication)
ISBN 978-9949-23-494-3 (PDF)

MEHHANOTEHNIKA E76

3D globaalkaardi ühendamise meetodid roboti navigeerimiseks

DMITRY SHVARTS

TTÜ
KIRJASTUS

Contents

Abbreviations 8
Introduction 9
  Background 9
  Simultaneous Localization and Mapping 10
  Visual SLAM 10
  From SLAM to co-SLAM 12
  The Scientific Objectives of the Thesis 15
  Structure of the Thesis 15
1. Review of existing methods 17
  1.1 Introduction 17
  1.2 Stochastic methods 17
  1.3 Likelihood method over robots' positions 18
  1.4 Landmark-based algorithms 19
  1.5 Visual appearance 19
  1.6 Conclusion 21
2. Use of global descriptors as an alternative to local descriptors. 22
  2.1 Introduction 22
  2.2 Scene structure 22
  2.3 Global descriptor for solving a loop–closing problem in SLAM 24
  2.4 Mathematical model of GIST representation of an image 26
  2.5 The GIST descriptor in loop-closing applications 30
    2.5.1 Image representation 31
    2.5.2 Image retrieval algorithm 32
  2.6 Experiment 34
  2.7 Conclusion 44
3. Map-merging problem in a multi-robot application 45
  3.1 Introduction 45
  3.2 Review of existing methods 46
  3.3 Artificial Neural Network 48
    3.3.1 Model of an artificial neuron 48
    3.3.2 Neural network architectures 49
    3.3.3 Approximation of nonlinear functions 50
    3.3.4 Multilayer perceptron 51
  3.4 Map-merging problem as a task of nonlinear function approximation 51
  3.5 Result of the experiment 56
  3.6 Conclusion 67
Conclusion 69
  Scientific Results 69
  The Main Scientific Contributions 70
  Future work 70
References 72
List of Publications 77
Abstract 78

Kokkuvõte..80
Elulookirjeldus..82
Curriculum Vitae 83

Acknowledgments

I would like to express my deep gratitude to my supervisor, professor Mart Tamre for his faith in myself. He supervised me, fully helped and supported through the process of learning. I would like to thank him for the knowledge and experience I received while working and learning under his competent leadership.

I would like to thank Administration/Management of Virumaa college for their understanding and support. Personal thanks to Victor Andreev, the Director of the College and Mare Roosileht, Director for Development.

I would like to thank my colleagues, Tatiana Barashkova, Assoc Professor, Genadi Aryasov, Assoc Professor for the time they spent with me in the discussions, for their advice and support. I would like to tank Irina Petrova, Natalja Denisova, Mare-Anne Laane for their help in English. Thank you!

I would like to thank Janne Heikkilä, Professor, from University of Oulu, Finland. He gave me the opportunity to work in the laboratory of computer vision for 4 months and introduced me to the world of SLAM technology.

I would like to thank Estonian Ministry of Education and Research (Project SF0140113Bs08 „Mechatronic and Production Systems Proactivity and Behavioural Models“) and European Social Fund (Project"Doctoral School of Energy and Geotechnology II”) Archimedes Foundation for financial support of this study.

I would like to thank my relatives and friends for their support.

This work is dedicated to my dearest people, my parents and my wife. Thank you for what you gave me that you are always close to me, for the support and faith in myself. You give me the strength to create!

Abbreviations

SLAM	Simultaneous Localization and Mapping
SfM	Structure from Motion
EKF	Extended Kalman Filter
MSER	Maximally Stable Extreme Region
SIFT	Scale-invariant feature transform
NN	Artificial Neural Network
PCA	Principal component analysis
WFT	Windows Fourier transformation
DFT	Discrete Fourier transformation
GIST	An abstract representation of the scene
KLT	Karhunen-Loeve Transformation
BOW	Bag-of-Words
ICP	Iterative Closest Point algorithm

Introduction

“Our nature consists in motion...” said the French physicist and philosopher Blaise Pascal. But how often do we ask ourselves, “What enables us to successfully move around in space? “We are able to plan a route to assess the complexity of the obstacles encountered on the way and successfully skirt them. Nature has given us an amazing gift that we call a word ”see”. ”Making a computer see” - first words in an annotation to the famous textbook on computer vision written by one of the pioneers in this field Olivier Faugeras. At the dawn of this discipline, many researchers were not aware of the complexity of the tasks, and considered them as relatively simple. Apparently, this is due to the fact that for a human seeing is an everyday, usual action. But the details of this process are hidden. Even today, modern science cannot explain many of the processes occurring in the human organism by obtaining, analyzing and processing images. Nevertheless, many scientists are trying to give the computer an opportunity to see.

In recent years, computer vision researchers have achieved outstanding success. Many theoretical achievements have found their practical application. This thesis deals in particular with a possibility of using video information for navigation of a group of mobile robots.

Background

The term navigation is commonly understood as a technology of computing an optimal route. Therefore, any mobile robot requires the ability to build maps of an unknown environment, localize itself on it and use this map for navigation. Navigation of a mobile robot may be split into two levels. The first level is called local navigation. Such navigation is based on the information received from all kind of sensors. To construct a map of an interesting area Simultaneous Localization and Mapping (SLAM) technology is widely used. The robot simultaneously builds a map and localizes itself on it in real-time mode.

The second level of navigation is usually called global navigation. This level implies the ability of effective navigation and planning of routes within the boundaries of the constructed map. However, in many practical applications, robot has to operate in different places, often without any prior knowledge about them at all. In such situations the robot is unable to navigate effectively. The key solution to this problem is a collaborative behavior of a group of robots. In cooperative multi-agent systems several agents make an attempt through their interaction to jointly solve tasks or to maximize utility.

We start with an introduction in SLAM technology. Then we move on to justify the advantages of collaborative behavior of a group of robots. The last part of this chapter covers the aim and contribution of this thesis.

Simultaneous Localization and Mapping

The movement of a mobile robot in an unknown space has become an integral part of its autonomous behavior. Simultaneous Localization and Mapping (SLAM) addresses two key problems in navigation: the estimation of robot position and representation of the surrounding environment.

SLAM has developed rapidly during the last decades. It has proposed various solutions for the indoor and outdoor environment. Initial investigations focused on the implementation of estimation - theoretical methods for mapping and localization. The results reported in [51] show that there must be a high degree of correlation between the estimates of point's position on the map. This correlation grows with every estimation step. A popular statistical tool for solving and analyzing physical time-varying systems corrupted by noise is EKF [26]. Such systems are represented as a single vector of states and information about the state is reflected in the multidimensional probability distribution. At each estimation step, the system carries out measurements and the results of these measurements are integrated into the probability distribution. EKF allows consistent and effective estimation of the vector of state. The computational complexity of the system depends on the size of the state vector. Such filtering methods of estimation of three-dimensional information have been used in SLAM. In [7] EKF based SLAM was implemented for a moving robot with ultrasonic sensors on board. In [39] a stereo rig was used as a sensor and EKF was designed to reduce triangulation errors.

From a theoretical point of view, SLAM seems to be a solved problem, but many issues of practical implementation require improvement. They are mainly aimed for reducing the computational complexity, data association and environment representation.

Visual SLAM

For an autonomous movement in an unfamiliar area, a robot has a wide range of distance sensors on board. They perceive the environment and supply data to construct a map. First SLAM implementations made use of multiple sensors often combined with odometry [1]. The main representative of such devices is a laser rangefinder. It combines a unique and unrivaled combination of a wide field-of-view, high maximum range, and fast data acquisition. However, a major disadvantage, which limits their use in many practical applications, is the high price.

In comparison, a camera-based system gives significantly more information about the environment than any kind of a distance sensor. From a single image, we can obtain information about thousands of points of the environment. Unique description of the points allows us to find these points in a database or in other images. Sensing range of the camera tends to infinity. Needless to say those for navigation purposes, far points, and a star for example, are useful like the nearest points. In the context of visual geometry, these points are called points of infinity. Such features cannot be used to estimate camera translation but they are a perfect bearing reference to estimate rotation [6].

Use of video data helps to overcome one of the greatest impediments to the long term and robust SLAM "loop closing" problem. Visual image-feathers are used in conjunction with scanning laser data [20]. Solving the SLAM problem with a vision sensor as the only external sensor is a key area in robotics research. Monocular vision is beneficial as it offers a very inexpensive solution.

Nowadays, one of the key elements of any robotic system is a machine vision system. It provides necessary knowledge of the environment. The 3D reconstruction is one of the important issues in any SLAM implementation.

Machine vision systems can be divided into two classes: Active Visual sensing and Passive Visual sensing. The possibility of using bothsystems, measurement principles, estimations of 3D information methods are described in detail in [5]. In active visual sensing techniques, an external device is used. It emits light patterns that are reflected by the scene and detected by a camera. Two measurement methods are employed in this class of visual sensing: triangulation or time-of-flight.

The other class of visual sensing is called passive sensing. Devices related to this class require no external projecting devices and use an ambient light for measurement only. This explains their low cost and attractiveness. Such passive techniques include stereovision, trinocular vision, and monocular vision.

However, the reconstruction of a 3D structure from a single 2D image is an Ill-posed problem. Methods and algorithms from projective geometry theory provide the possibility to compute a 3D structure of a scene.

Structure computation is a line of research, which allows reconstructing of 3D information from a set of 2D corresponding points. A couple of decades ago, this challenging problem offered a wide field for research. Today this problem could be solved in two ways. One possibility is realized by reduction of degrees of freedom and through the use of additional constraints (for example, the distance from the camera to the image plane). However, with such assumptions a scene cannot be reconstructed in an automatic mode.

The use of a set of images is another possibility to solve the reconstruction problem. The classical tool in the reconstruction is a fundamental matrix. This matrix represents all geometric relations which exist between two 2D images. The corresponding points detection process was carried out by hand. A pair of matching points x and x' and matrix F must satisfy

$$xFx' = 0$$

However, in practice, all measurements of matched points are corrupted by noise, i.e. measured points will not satisfy the epipolar constraint $xFx' = 0$ and it will be impossible to estimate 3D coordinates of a space point X. Finally, a nonlinear optimization method was added to the processing mechanism to minimize a suitable cost function.

The need for the automation of existing reconstruction methods has activated the researcher to develop new robust methods for finding interesting points and lines, as well as full automation methods for computing matched points on different views.

The result of these studies is the invention of a number of automation methods for scene reconstruction and camera pose estimation. The foundation for these methods was a theory of classical structure from the motion problem (SfM), which was supplemented with some kind of nonlinear optimization methods [59].

From SLAM to co-SLAM

Recent developments of robots with autonomous behavior are substantial, but practical implementation needs considerable improvement. A new area of research into the benefits of group behavior has emerged. In such systems, focus is on the use of the collective capacity of a group. Before examining the feasibility of using systems with group behavior for solving navigation tasks, we will introduce a few concepts. The term “system with group behavior” commonly refers to a set of subsystems without centralized control that performs similar functions. The phenomenon associated with the transition from chaotic behavior to the target group behavior is called a phenomenon of self-organization. In [32] the main advantages of the systems with group behavior are presented. We highlight only some of them:

Systems without centralized control are cheap, resulting from simple, single-type devices used in the production and maintenance of such systems.

Self-organizing systems allow creating complex functional systems by a set of simple elements. The group behavior in such systems is carried out in the interests of the entire colony. Joint work of bees to maintain a constant temperature of $+32^{o}$ C in the hive is an example of the group behavior in biological systems [32]. Self-organization in artificial systems can be applied to solve many problems associated with exploring of a terrain, searching for resources. "Swarm robotics project" is a bright example of artificial self-organizing systems. In a group of robots endowed with the ability to collectively solve such problems, the survivability of not only individual elements of the system but of the entire colony will increase.

Self-organizing systems are invariant to the number of elements of the system. The number of elements of the system will not change its functionality. In the event of failure of one element, the performance of the whole system is preserved. This fact is particularly interesting. Self-organizing systems can be used successfully both in large enterprises and small companies. For example, originally a robotic system for a warehouse had acquired a centralized management and was very expensive. The use of such a system in small companies was economically infeasible. Implementation of a self-organization technology makes these applications accessible to a wide range of consumers.

The element of the system with group behavior is an agent. In general, an agent is hardware or (more commonly) a software computer-based system capable of acting to attain the goals stated by a user. In this thesis, an agent is defined as a mobile robot that has a certain set of properties as in [46]:

Autonomy:	An agent works as an independent, autonomous vehicle that sets goals and acts to attain these goals.
Social ability:	An agent can communicate with other agents.
Mobility:	The ability to transfer the agent data to a server or to other agents.

A group of such agents builds a multi-agent system. Due to interaction between the agents, the multi-agent problem complexity can rise rapidly. Multi-agent systems may consist of different kinds of robots. They can differ not only in hardware or software but in their behavior and tasks they perform.

Current algorithms for solving mapping and localization problems have been implemented for single robots. Collaborative behavior has several advantages. A group of robots can solve the problem of mapping of the environment and completing a task or work more quickly and robustly than a single robot.

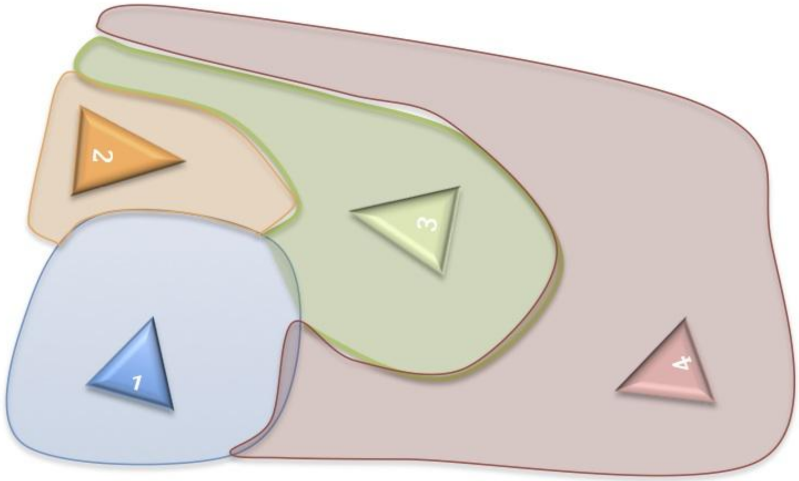


Figure I.1. An example of a simple scenario

Figure I.1 presents a simple example scheme of using self-organization to solve problems of autonomous movement in an unknown environment. Each robot exploring a space simultaneously builds its own map. For example, a robot 1 has explored and built a map of the green area. It should be emphasized that effective robot path planning is possible only within the boundaries of the map due to the absence of necessary information from outside area. An effective solution to this problem is possible by using the missing part of the map, which was created by a neighboring robot as a result of mutual concerted action. To use these maps together it is essential to merge them accurately and, what is the most important for this kind of robot behavior, in real-time consuming less computational power, which will allow the algorithm to run onboard of the robots.

An example of the practical application of such scenario is shown in Figure I.2. A colony of industrial robot platforms, without any central control system able to perform logistic tasks in autonomous mode. The system's ability to automatically adapt to any environment, make it available for both large and small warehouses [34].

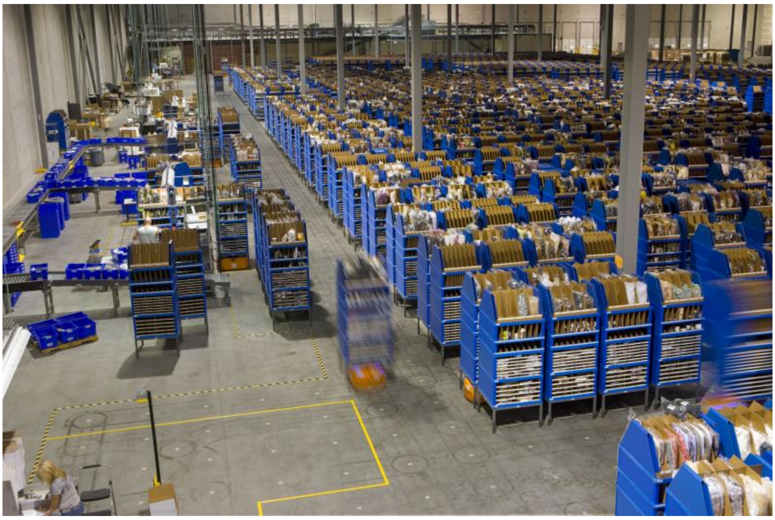


Figure I.2. The colony of warehouse robots

Other possible applications are presented in Figure I.3.

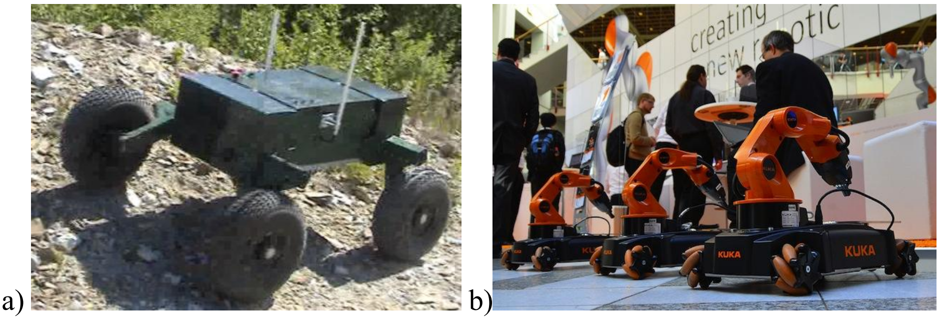


Figure I.3. a) Security patrol robot from TUT may share a map with other robots [56], b) Industrial robots can work on industry sharing workplace information and maps [17].

Despite the fact that the study of the individual "SLAM" has made great progress, group behavior, as the key to greater autonomy, is insufficiently studied. Due to the problem that the SLAM realizations depend greatly on the environment conditions, i.e. indoor or outdoor, the following research has been limited to indoor conditions although the approach is generally applicable for any conditions.

The Scientific Objectives of the Thesis

The Scientific Objectives of the thesis:

- Development of an intelligent method for solving map-merging problem.
- Development of a new algorithm for use the method on robots in indoor conditions.
- Development of a reliable, efficient algorithm for detection overlaps of existing maps based on visual appearance.
- Development of a reliable, efficient algorithm for aligning two 3D local maps in a common coordinate system.

The aim of this work is to develop a reliable method for merging several local maps in a global map. This method will enable a step-up from the local autonomous robot navigation toward global navigation, toward collaboration of groups of robots for solving more complex problems. A visual sensor is considered as a sole sensor and applied purposively and selectively to acquire and use data.

Structure of the Thesis

Multi-robot systems have several advantages over a single agent case. Using the collective potential, a group of mobile robots can solve different tasks more efficiently, including those that a single robot cannot solve independently. Multi-robot systems are robust and invariant to the number of agents in the system. The use of the collective capacity allows for reducing requirements to the technical capabilities of a single robot. Thus, each robot becomes interchangeable, cheap to produce and the system technically survivable, reliable and easy to install. Analysis of these advantages certainly encourages us to develop new methods of interaction between individual robots in a multi-robot system by solving a map- merging problem. To reduce the production cost of an individual robot the single camera as a sensor was used. Besides low production cost, the camera provides additional information about the surrounding environment.

The study consists of four parts:

Chapter 1 – Review of existing methods: Existing methods for the map-merging system are briefly considered; major advantages and disadvantages of the presented methods are highlighted.

Chapter 2 – Use of global descriptors as an alternative to local descriptors: The map-merging problem could be split into several simple tasks. Some of them are partly solved in a single robot case. One of them is the loop-closing problem. A novel technique for solving the loop-closing problem is presented. The proposed method enables a more efficient solution of the loop-closing problem in a single robot case. The advantage of this method is in its capability to extend it to a multi-robot case and successful solution of the map-merging problem. This strategy may be also useful in application for place or object recognition.

Chapter 3 – Solution of map-merging problem in a multi-robot application: The author offers to use the capacity of the neural network in map-merging applications. Some assumptions proposed by the author help to present this issue in more general terms.

Conclusion: Generalizations and ideas for future work are presented.

As stated in the introduction, the main purpose of the thesis is to develop a reliable method of combining several local maps created by mobile robots in a single global map.

To develop this method, first, analysis of the state of art in this field is required. The next chapter reviews the existing methods.

1. Review of existing methods

1.1 Introduction

In the last decade, focus of research in autonomous robotics has been on developing and refining the algorithms of autonomous agent behavior. Many tasks have been solved. However, in practical applications the robot should be able to solve problems in collaboration with other robots. In fact, the speed and quality of a solution depends on the ability of robots to work in a team. Undoubtedly, one of such problems is the navigation.

Existing algorithms for group behavior of robots for solving a map-merging problem can be broadly divided into three main classes [20]:

- Stochastic methods: merging sensory data from multiple robots with known data association between features in local maps built by different robots.
- Likelihood method over robots' positions: detecting other robots to determine relative position and orientation between local maps or assuming relative poses between robots are known.
- Landmark-based algorithms: deriving the transformation between robots' coordinate systems through the matching of landmarks.

This chapter covers the essence of each class and reveals their advantages and disadvantages. The state of the art in this field and the main direction of work will be presented.

1.2 Stochastic methods

The first group of SLAM algorithms has used the stochastic methods. Algorithms for cooperative behavior were an expansion of the "single SLAM" problem to the multi-robots case. In [12] the system of a group of robots was represented by a single combined state vector.

$$x[k] = \begin{bmatrix} x_v[k] \\ x_l[k] \end{bmatrix} = \begin{bmatrix} x_v^A[k] \\ x_v^B[k] \\ \vdots \\ x_v^N[k] \\ x_{l_1}[k] \\ x_{l_2}[k] \\ \vdots \\ x_n[k] \end{bmatrix} \tag{1.1}$$

Collaborating vehicle state estimates $x_v[k]$

$$x_v[k] = [x_v^A[k]^T x_v^B[k]^T \dots x\,{}_v^N[k]^T], \tag{1.2}$$

where $x_v^i[k]^T$ is the vehicle state estimate for a vehicle i at a time k.

The state estimate of the i^{th} landmark at time step k is represented by the position estimate $x_{l_i}[k]$ and is included into the vector state of the environment

$$x_l[k] \ = (x_{l_1}[k], x_{l_2}[k], \dots, x_{l_i}[k], \dots, x_{l_n}[k]). \tag{1.3}$$

A single state estimate of the represented system incorporates all of the vehicle and feature estimates. On each estimated step, the vector state estimate and associated covariance matrix P of the estimation error are updated using an Extended Kalman Filter (EKF).

$$P = \begin{bmatrix} P^{AA} & P^{AB} & \cdots & P^{AN} & P^{A1} & P^{A2} & \cdots & P^{An} \\ P^{BA} & P^{BB} & \cdots & P^{BN} & P^{B1} & P^{B2} & \cdots & P^{Bn} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ P^{NA} & P^{NB} & \cdots & P^{NN} & P^{N1} & P^{N2} & \cdots & P^{Nn} \\ P^{1A} & P^{1B} & \cdots & P^{1N} & P^{11} & P^{12} & \cdots & P^{1n} \\ P^{2A} & P^{2B} & \cdots & P^{2N} & P^{21} & P^{22} & \cdots & P^{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P^{nA} & P^{nB} & \cdots & P^{nN} & P^{n1} & P^{n2} & \cdots & P^{nn} \end{bmatrix}. \tag{1.4}$$

This method has a number of disadvantages common to all algorithms based on the filtering method. The main drawback is that the graph of a system's state becomes fully inter-connected [54]. This leads to an increase in the computational cost. It is well known that computational cost of filtering methods is $O(n^2)$, where n is the total numbers of landmark in the map. However, in [54] it was shown that in some cases the computational cost can grow to $O(n^3)$.

1.3 Likelihood method over robots' positions

The key point of the second group of methods is the detection of position and orientation between two robots. The position of any mobile robot is uniquely defined in the fixed coordinate system of a local map. Using the information about the relative positions of robots, the task of map merging becomes feasible. In [31] the author formulates the map-merging problem as a decision problem in terms of likelihood over the position of the robot. The key aspect of solving the decision problem is using features with good discrimination power. A probabilistic approach to collaborative multi-robot localization was proposed in [13]. In this work, an implementation that uses a color camera and laser range finders to determine a robot's position is presented. Other studies have used some assumptions, for example, relative poses between robots are known. Using the method outlined above, a situation may arise when two robots explore the same environment without being aware of each relative position and orientation. It is the main drawback of the second group of methods.

1.4 Landmark-based algorithms

New ideas to solving the map-merging problem appeared in [9]. This paper describes a landmark-based algorithm for map matching. The transformation between robots' coordinate systems is derived through the matching of landmarks. In short, at the map matching steps the pair of matching points (l_i^{R1}, l_i^{R2}) is tested for possible mismatches. For all remaining candidate pairs the transformation components are computed. This technique can be applied to all other pair candidates and counting of overlapping and matching landmarks. Having tried all possible transformations, the winner is the transformation with the highest number of match-counts, also called the best match.

The map-merging approach proposed in [20] uses visually salient features for local maps intersection detection. According to this work, the detected "similarity" can be later used for maps alignment. Despite the fact that the proposed method shows a good result, there are a couple of disadvantages that limit its use in practical applications. This method will be discussed in detail in the next chapter.

1.5 Visual appearance

The advantages derived from using of video cameras in SLAM applications allowed us to consider the camera as a major sensor. With equal working time for exploring an environment, the volume of received information has far exceeded the previous value. As a result, in a number of works the map-merging problem has been solved by using visual appearance. A novel method was presented in [20]. This method, the starting point of the thesis, is discussed in detail in this chapter.

A group of robots starts to explore the environment from different points. Each robot builds its own map by using video SLAM technology. The map-merging process is based on the comparison of two image sequences. Each image is described by a set of visual points. In contrast to previous methods, the MSER visual point's detector has been applied [38]. Each new image feature extracted by the MSER detector was coded using the popular SIFT descriptor [36]. The MSER detector offers higher efficiency and higher speed of the detection of a subset of visual features with the stability to the affine transformation. The SIFT descriptor raised discriminative properties of such points. Then, the calculated weight is assigned to different SIFT descriptors based on the frequency of their occurrence in the image database. A method of building a database is known as bag of words (BOW), first presented in the computer vision community in [37]. As a result, each image is represented by a vector $I_u = [u_1 \cdots u_n]$, where u_n is a weighted descriptor or a visual word. The visual similarity matrix M constructed from such comparison between two image sequences is collected by different robots.

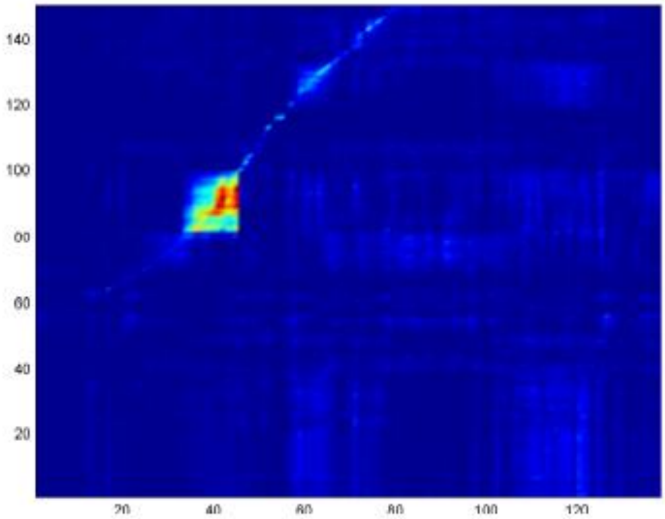


Figure 1.1 . Visual similarity matrix. The bright line highlights the sequence of images that are similar to each other – indicating that there is an overlap in the two environments explored. This image was taken from [20].

Entities of the matrix $M_{i,j}$ are the estimates of visual similarity between the two images. Since any image is represented as a vector, the similarity between the two images can be easily derived by using the Euclidean dot product formula:

$$S(I_u, I_v) = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{(\sum_{i=1}^{n} u_i^2)^{1/2} \cdot (\sum_{i=1}^{n} u_i^2)^{1/2}}, \qquad (1.5)$$

where $I_u = [u_1 \cdots u_n]$, $I_v = [v_1 \cdots v_n]$ are the representations of images in the BOW style.

This method tested in the indoor environment at the distance of 180 meters has shown a good result. However, the existing shortcomings limit the application of this method in practice. A visual similarity matrix is constructed from a comparison between two images sequences collected by two robots. The time required to compare the type "with each other" by constructing the similarity matrix is highly dependent on the amount of images in the database of both robots and also on the dimension of visual dictionary. But an obvious disadvantage of the BOW method is that it is time consuming. The use of long sequences of images greatly increases the time for constructing a visual vocabulary. As shown in Figure 1.1, the sequence of 146 images (y-axis) for constructing a visual dictionary is used. By increasing the distance traversed by the robot increases the number of collected images, respectively. If the number of images reaches 4000 - 6000, the calculation time may increase up to several hours. Because of the shortcomings, this method needs detailed research.

1.6 Conclusion

This chapter has presented an overview of the existing methods for solving the map-merging problem. The analysis has confirmed the theoretical validity of those methods, but also helped to identify a number of drawbacks. Highlighted shortcomings limit the use of these methods in practical applications. The group of landmark-based algorithms deserves special attention. Algorithms of this class make use of only available data that a robot creates during the execution of an individual SLAM. The most successful example of this class of algorithms is the algorithm that uses visual appearance for detecting “similar” intersections between local maps built by multiple robots. The availability of effective techniques for the detection of visually salient features makes this kind of algorithm more attractive. A particular advantage of landmark-based algorithms is the lack of need of additional equipment. Finally, the multiple map intersection detection using the visual appearance algorithm was examined. The obvious drawback of the method is the image search time dependent on the number of images in the database. The query image is compared against every image in the database. The image retrieval time is increased due to increased number of images in the database. Such drawback limits the use of this image search strategy in practical applications. Along with these shortcomings, the method has several advantages. Each image is described by a set of local features, and this set will use the latter to detect similarity between the two images. In image retrieval systems, which are based on the principle of verification of local matching, the number of false positives is significantly reduced. The detected strengths and weaknesses of the method helped to shape the future direction of research. It became apparent that the task of aligning of several maps can be roughly divided into three stages:

- Every robot carries out the individual SLAM technique and builds a map of the environment explored.
- Image database is developed through accumulation of images from several robots; the query images are searched in the database.
- If the system detects an image similar to query imagery, the robot will explore the area that has already been explored by another robot. It is only necessary to calculate the coefficients of the transformation matrix to transform the 3D coordinates of the existing map in the coordinate system of the requested robot. Thus, the two local maps are combined in a global map.

The next chapter describes the properties of global descriptors and introduces a new approach to use them more effectively. We compare an image retrieval system based on global and local descriptors. A new image retrieval algorithm that uses global and local features of the scene is presented.

2. Use of global descriptors as an alternative to local descriptors.

2.1 Introduction

To construct a global map of the explored environment, robot group behavior could be roughly divided into three steps:

- Constructing local maps
- Finding of existing overlapped areas
- Merging several local maps in a global map

The first step, the local map building process, is a subject of an individual SLAM. It must be noted that the visual SLAM applications have shown some benefits and a camera is considered in this thesis as the only sensor.

During the second step, some tasks that a robot solves during the SLAM process are very similar to the tasks for solving the map-merging problem, for example, a loop-closing problem is comparable to the problem of finding overlapped maps. Traditionally, in a visual SLAM, a loop-closing problem is solved by means of local structure analysis. Each image is represented as a frequency of occurrences of local features. This method is well known as the bag-of-words method. However, the image searching strategy based on the BOW method has two limitations: complexity and memory usage. Some extensions were applied to the BOW method to improve its properties. The BOW representations of images have shown very good results in image search, but the problem of memory usage is still unsolved. Parallel to this, in the computer vision literature, global descriptors have received increasing attention, including the GIST descriptor. This descriptor is very fast and compact. But due to the well-known limitation of global descriptors, as they are not invariant to significant transformation and lighting, global descriptors are not exploited in visual SLAM applications.

The third step is described in detail in the fourth part.

In this chapter we evaluate the image search strategy and increase the search accuracy of the system based on a global descriptor for solving a loop–closing problem in a visual SLAM application.

2.2 Scene structure

The image searching process in a large database using the image content is an interesting and challenging task. Principles of operation of existing systems are similar in many ways. In a database the system search for an image or group of images is visually similar to the query image. The degree of similarity of the query and matched images directly depends both on the properties of the system and on the application problems in whose interests this searching is carried out. Obviously, the degree of similarity of images should be maximum if the matched image should be used for a task like 3D reconstruction of the scene or place recognition. In [45], the author has identified three levels of scene description. Such a division determines the necessary level of detailization, the presentation of the requested image and applied mathematics.

The first level, the **subordinate level**, is a level of detailed description of the local area of the image. This level is often used in place recognition applications, for the analysis of local structures of an image. Chad Carson has proposed an image search method called "Blobworld". The Blobworld image representation involves three steps (see Fig. 2.1.):

1. Extract color, texture and position features for each pixel at the selected scale.
2. Group pixels into regions.
3. Describe the color distribution and texture of each region for use in the query.

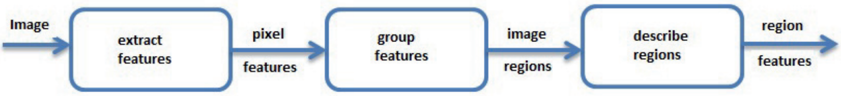


Figure 2.1. From pixel to region description.

Modern methods for image representation use simpler algorithms, which still have large discriminatory properties. A common strategy for image representation is shown in Figure 2.2.

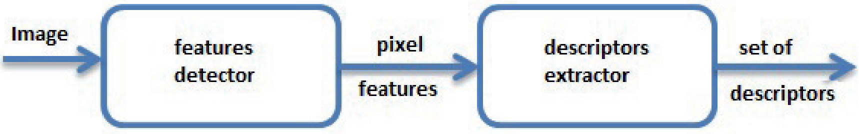


Figure 2.2. Image representation with local features.

A search system returns an image selected on the principle of texture, color or spatial structure similarity.

The second level is called the **basic level,** characterized by the general description of the scene. Roughly, it subdivides a scene on artificial (houses, streets) and natural landscape (forest, mounts). Images referred to a group usually involve similar objects.

The third level, called the **superordinate level**, usually characterizes the scene with a maximum degree of abstractness. Images are classified into artificial and natural landscapes made inside and outside. A more general set of categories is proposed in [60]. A scene may refer to one of the categories (inside, outside, city landscape, wildlife, mountains, and forest).

In many SLAM applications, the degree of image representation belongs to the first level and is carried out using low-level visual features. Analysis of local structures is needed to define re-visited places, precise localization of the robot or to reconstruct the 3D scene. However, this level presents images as high dimensional vectors and this fact complicates processing, storage and retrieval of images.

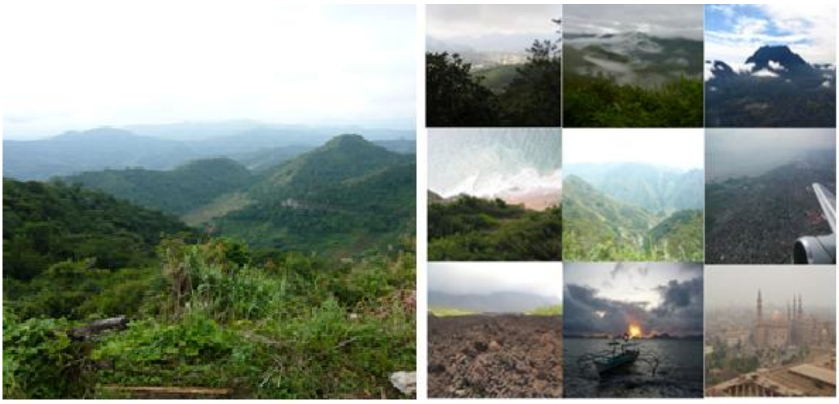
2.3 Global descriptor for solving a loop–closing problem in SLAM

Global features have been used in the computer vision community as an alternative to local features for scene classification [45], [60]. Their key advantage for this application is that their performance is very similar to that of local features at a much lower cost [11]. In recent years, in view of global features, greatest interest has been shown in the robotics community. Most shape and texture descriptors may belong to this category. Such features are attractive because they produce very compact representations of images, where each image corresponds to a point in a high dimensional feature space. For a global descriptor, in this thesis we will use the popular GIST [45]. Previous work has shown that the global descriptor may be successfully applied for scene classifications represented on the base level or on the subordinate level of scene description. The results of experiments confirm the validity of this claim. Query image represented as a high-dimensional vector is compared against all images from the database represented in the form of high-dimensional vectors too. In the experiment and in the further work, this thesis uses the popular GIST descriptor. Minimum Euclidean distance between the compared GIST descriptors, meaning the query image and its likely candidate, determines the degree of similarity of images.

$$\begin{aligned}\rho(x_n, y_n) &= \|x - y\| = \\ &= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \\ &= \sqrt{\textstyle\sum_{k=1}^{n}(x_k - y_k)^2},\end{aligned} \tag{2.1}$$

where $x = \{x_1, x_2, \cdots, x_n\}$, $y = \{y_1, y_2, \cdots, y_n\}$.

Figure 2.3 shows the query image and its nine scenes nearest candidates in the sense of minimum Euclidian distance.

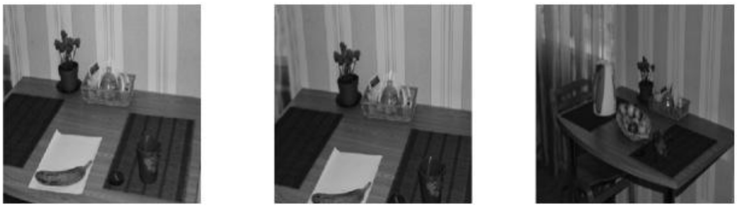


a) Query image

b) nearest candidats

Figure 2.3. Result of the experiment; a) query image and b) nearest candidates.

The figure shows that global features representation of the image reflects the general structure of the scene, at the same time completely ignoring the local structures. From the scene classification point of view, the minimum Euclidean distance has the images that belong to the same group by the basic or superordinate level of scene presentation. But the problem of loop closing greatly differs from the scene classification one; as the aim of loop closing is to discern if we are in the same place and that of the scene classification is to determine whether a scene belongs to the same category. Based on this observation we could suggest that global descriptors are completely unsuitable for the determination of re-visited places. However, recent studies suggest using global features in the robotics context. For example, in [35] proposes a SLAM system based on particle filters that select loop closure candidates based on GIST features. In [43] the GIST descriptor for panoramic images is also used. In [11] global features are successfully used for web searching in databases with millions of images. During the experiments it was observed that using the global descriptor can most likely determine the exact same scene in the two images when the images were taken from approximately the same location. On the one hand, it confirms one of the drawbacks of the global features that they are not invariant to affine transformations, but on the other hand, it highlights more clearly the properties of the global descriptor. Even if the details of the scene have changed, the global structure of the scene remains unchanged. It allowed identifying the scene as the same. The results of the experiment are presented in image 2.4.



Query Image

Image 1
ρ = 0.33

Image 2
ρ = 0.6456

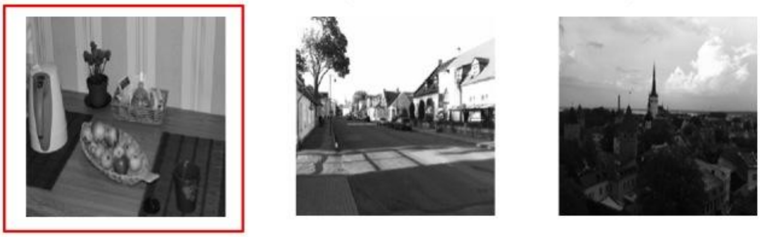


Image 3
ρ = 0.4040

Image 4
ρ = 0.6699

Image 5
ρ = 0.7362

Figure 2.4. Compared images.

2.4 Mathematical model of GIST representation of an image

As mentioned in previous chapters, a GIST descriptor encodes the global information about a scene and remains indifferent for local changes. Such unique property of the GIST descriptor reflects the global structure of the scene, explained by the nature of the formation of GIST representation, which is the basis of the discrete Fourier transform. In fact, any grayscale image is a 2D function $f(x,y)$ (digital image), where x and y are spatial coordinates and magnitude f is intensity of the image in these coordinates. If $f(x,y)$where $x=1,2,3,\dots,M$ and $y=1,2,3,\dots,N$ image with size $M\times N$, then, the two-dimensional Fourier transformation $F(u,v)$ of the image f can be expressed by the equation:

$$F(u,v) = \sum_{x=1}^{M}\sum_{y=1}^{N} f(x,y)e^{-j2\pi\left(\frac{xu}{M}+\frac{yv}{N}\right)}, \tag{2.2}$$

where $u=1,2,3,\dots,M$ and $v=1,2,3,\dots,N$ – frequency variable.

Despite the fact that the magnitude of the $f(x,y)$ function belongs to the domain of real numbers, the value of a two-dimensional Fourier transform is the complex numbers $F(u,v)=R(u,v)+I(u,v)$, where $R(u,v)$ is the real and $I(u,v)$ is the imaginary part. In practice, to visualize the DFT $F\,(u,v)$ its amplitude spectrum is often used:

$$|F(u,v)| = [R^2(u,v)+I^2(u,v)]^{1/2}, \tag{2.3}$$

and the phase function of the DFT:

$$\Phi(u,v) = \arctan\left[\frac{I(u,v)}{R(u,v)}\right]. \tag{2.4}$$

In [45] it is shown that the phase function reflects information of local structures of the scene, and the amplitude spectrum of the image reflects the global structure of the scene. The amplitude spectrum reflects the frequency of brightness changing distributed over the whole image. Values of amplitude spectrum depend on the geometric dimensions of contours, the length and width, their smoothness. The forms of the amplitude spectrum of very simple objects are shown in Figure 2.5. The value of $F\,(0,0)$ placed in the middle of the image determines the "dc" frequency of the amplitude spectrum and corresponds to the average brightness of the image. As seen, visualization of the amplitude spectrum is clearly related to the orientation of the rectangle on the image. In Figure 2.5 b the white rectangle is located along the y-axis, but the maximum of the amplitude of the spectrum is arranged along the x-axis that is rotated 90 degrees relative to the orientation of the rectangle. We can see also that there are certain values of the amplitude spectrum along the y-axis but the magnitude of these values is lower. Another important issue is the distance between zero points of the amplitude spectrum. The distance along the u-axis is larger than the respective distance along the v-axis and this ratio depends on the ratio of the lengths of the sides of the rectangle. Figure 2.6 shows this relationship in detail. This is explained by the difference in the ratio of the length and width of the rectangle.

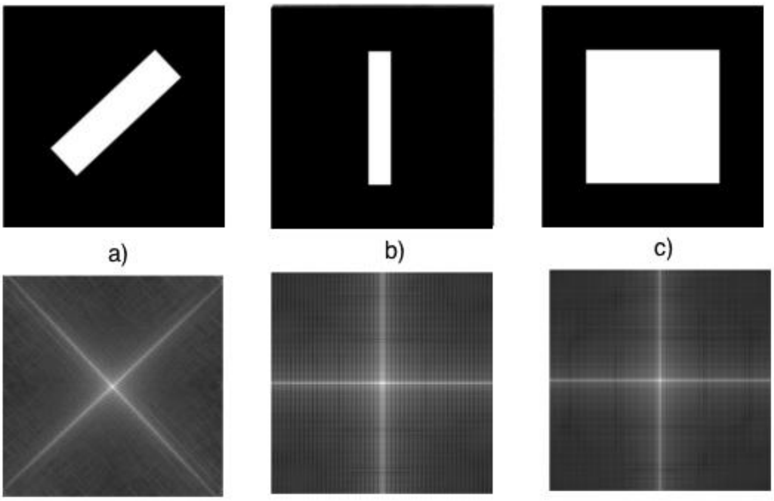


Figure 2.5. Amplitude spectrum of very simple forms.

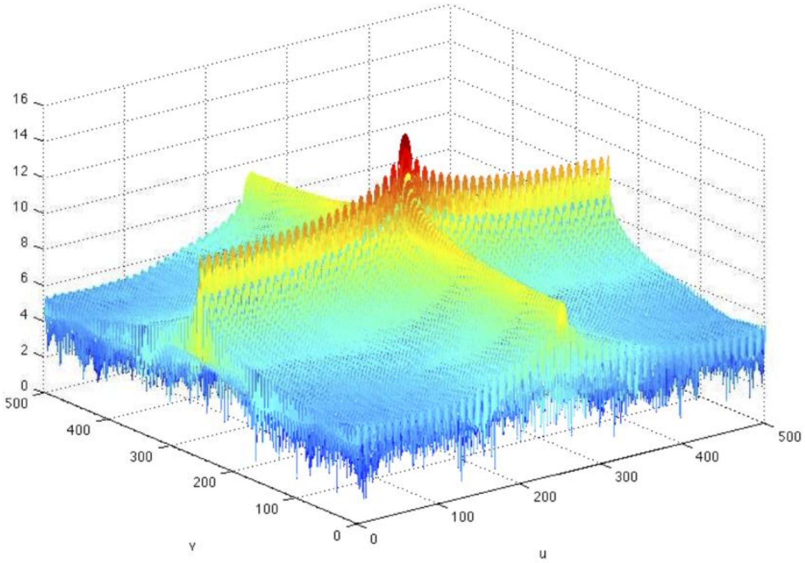


Figure 2.6. 3D representation of amplitude spectrum of the white rectangle located along the y-axis

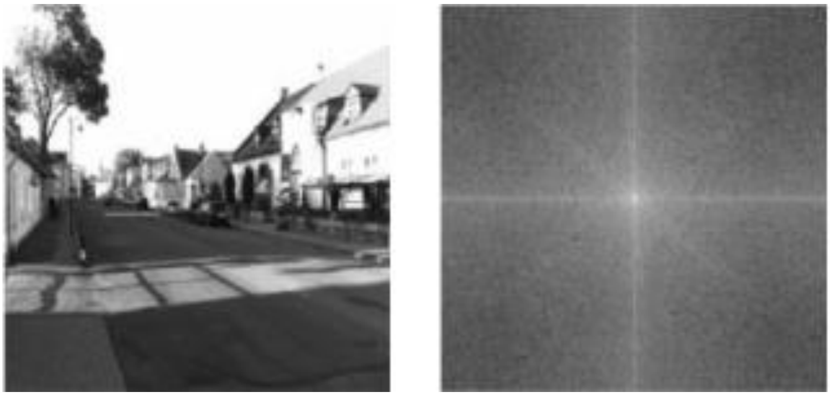


Figure 2.7. An image of a city landscape and visualization of its amplitude spectrum

In more general terms, the amplitude spectrum is closely related to the speed of brightness changing on the image. The low frequencies correspond to a smooth change of the brightness. The high frequencies correspond to the quick change of the brightness, for example, the presence of noise. The middle frequencies are a subject of edges of objects and its parts. Obviously, the Fourier transform $F\ (u, v)$ involves all of the f value of the function $f(x, y)$ multiplied by the exponential part (2.2). Thus it is usually difficult to establish a correspondence between the parts of the image and its spectrum, except in very simple cases, as presented in Figure 2.5. Figure 2.7 shows an image of the city landscape and its visualization of amplitude spectrum. In the picture, objects with vertical orientation (houses, trees, lamp posts) and with horizontal orientation (sunlight on the road) are dominating, but it is impassable to identify each particular object on the visualization of the spectrum image. Figure 2.8 presents more images with natural artificial landscape. Each image has a pronounced boundary of brightness changing, which is reflected in its Fourier transform. Along with amplitude spectrum, another representation of DFT, called the energy spectrum, is often used.

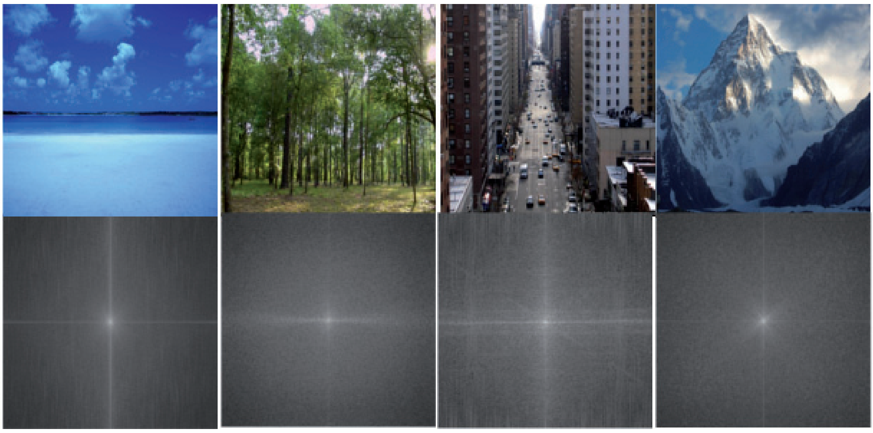


Figure 2.8. Images of natural and artificial landscape and its visualization of DFT

Energy spectrum $E(u,v)$ yields an idea of the distribution of the signal energy in the frequency domain and could be calculated by the following equation:

$$E(u,v) = F(u,v)^2. \tag{2.5}$$

As the amplitude spectrum and the energy spectrum are invariant to local changes of the scene elements, it encodes only the dominant structure of the scene. Since the dimension of DFT is equal to the size of the image $E(u,v) = N^2$, direct using of a raw result of the Fourier transform in many practical application is time consuming. Theoretically, in order to reduce the dimension of the function (amplitude or energy spectrum) Karhunen-Loeve Transformation (KLT, also known as eigenvector transform) and PCA can be used. The theoretical aspect of applying of KLT is explained in [45]. In fact, basic functions of the KL transform are eigenvectors of the covariance matrix of the input signal. This transformation is optimal for achieving the decorrelation criterion of an input signal and ensures that the percentage of energy in a given amount of the highest coefficients will not be less than in the same amount of coefficients by any other transformation. Often, however, there are difficulties in using this transformation in practice in computer vision applications. Due to the fact that in practice the number of images in a database is smaller than the number of elements in an image, the reliable estimation of basic functions is not possible. Authors have proposed another method to reduce the dimension of the DFT:

$$g_i = \iint E(u,v) G_i(u,v) du dv , \tag{2.6}$$

where $G_i$is a set of Gaussian functions arranged in a log-polar array and calculated by rotating and scaling the function:

$$G_i(u,v) = e^{-v^2/\sigma_v^2}\left(e^{-(u-f_0)^2/\sigma_u^2} + e^{(u+f_0)^2/\sigma_u^2}\right).$$

A vector g with dimensionality $g = \{g_i\}_{i=1,L}$ represents the function of energy spectrum $E(u,v)$. PCA is then applied to the low dimensional vector g.

For a detailed analysis of the scene in practical applications, the windows Fourier transformation (WFT) is often used. In this case, a grid splits the image in several image parts (spatial locations) and the energy spectrum is computed. Although this increases the dimension of the vector, at the same time it fills global scene representation with more unique information.

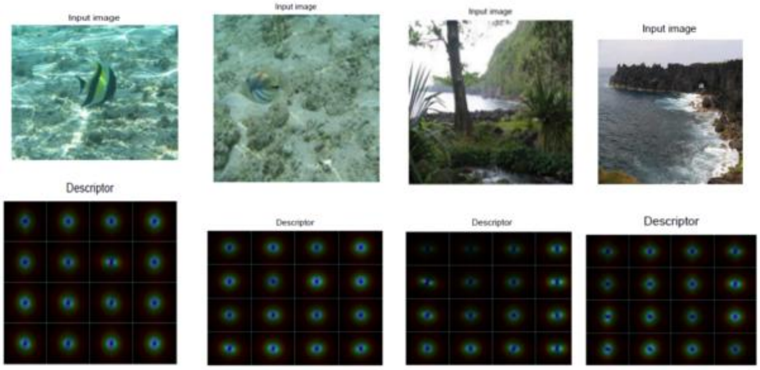


Figure 2.9. Several real images with corresponding GIST descriptor representations. GIST descriptor was computed by the windowed Fourier transform (WFT) at $\mathbf{4 \times 4}$ *spatial location.*

2.5 The GIST descriptor in loop-closing applications

The main task of any video SLAM algorithm is to build a map by using video information acquired by robots’ on-board cameras. Such maps consist of a collection of 3D points located in the world coordinate frame. Each point of the map has a geometric relationship with its corresponding source in the two-dimensional image plane. Such 2D points are usually called local features. Due the tracking of local features the robot is able to estimate its own position and continuously improve the map. In this context, the image matching process is a key aspect in addressing issues, such as object or scene recognition, solving 3D structure from multiple images, motion tracking. For reliable tracking, these points have to have a unique representation and a number of properties. They are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. Local features can be reliably detected and computed by a number of efficient algorithms. As shown in previous chapters, global features, and specifically, the GIST descriptor is widely used in computer vision applications. In this section the global descriptor for a specific purpose, recognition of re-visited places is used. The new efficient technique should increase the performance of any SLAM algorithm.

2.5.1 Image representation

The main task that a robot can solve by any SLAM algorithm is building a 3D map. In this context, the problem of detecting revisited places has acquired increased significance. The robot stores the representation of each frame in its memory, creates an image database and continuously checks the newly acquired image with elements of image database. In order to know if two images match, the distance between their bag-of-words models is computed. As the image descriptor is a histogram, the Bhattacharyya's distance would be the best option. If the distance is under a predefined threshold, the images are said to match. The images potentially matching can be geometrically verified using the image position of the local feature. However, previous analyses of image retrieval systems show that there are a number of limitations that degrade the properties of such systems, for instance, speed of search, computational complexity. To improve the performance of image retrieval algorithms we include its global feature in the image representation structure. At the same time, it is impossible to completely eliminate the use of the local descriptors. They are needed both at the stage of mapping and at the stage of image retrieving. The new image representation is a structure consisting of a global feature, in our case the GIST descriptor and a set of local features. The local features are the set of widely used SIFT features. The number of SIFT descriptors depends on the image content and can vary between 50 and 1350 local features in one particular image. Such number of local points increases the amount of memory consumed by each image. In addition, many of those local features cannot be used in further calculations. Therefore, to reduce the used memory, local features are computed in the following manner. MSER algorithm [10] detects a group of local points with specific properties. Then the SIFT algorithm encodes these features and presents each of them as a 128-dimensional vector. The final structure of the image is represented in the following formula:

$$\text{structure Frame} = \begin{Bmatrix} \text{global} = \{g_1, g_2, \dots, g_i\} \\ \text{local} = \{l_1, l_2, \dots, l_n\} \\ \text{bow} = \{p_1, p_{2,} \dots, p_k\} \end{Bmatrix}, \tag{2.7}$$

where global is a dimensional GIST vector, local is a set of 128-dimensional vectors of local features, and finally, bow is a bag-of–words image representation.

The dimension of the global representation depends on several factors, such as the size of the original image, the properties of the transformation algorithm from an image to its global presentation. The original image is split into a group of images with different scale and orientations. In our case, we have reduced the size of the original image to 32 × 32 pixels and applied 3 scalings and 8 orientations. The output value of the energy spectrum averages by 16 non-overlapping windows. As a result, the input image is represented as a $3 \times 8 \times 16 = 384$ -dimensional vector. Finally, the average amount of memory occupied by such representation is $4\,\text{bytes} \times 384 - \text{dimensional} = 1536\,\text{bytes}$ for gist features and $L \times (4\,\text{bytes} \times 128 - \text{dimensional}) = L \times 512\,\text{bytes}$ for local features. The experimental result shows that the average amount of the occupied memory of real images from the RAWSEED dataset is approximately 26K.

2.5.2 Image retrieval algorithm

This part of the chapter offers a new strategy for recognition of re-visited places. Such image search strategy was first presented in [11]. The main difference between the two methods is that the robot does not store all images in its memory but has only the structural representation, as described in the previous section. It should be noted that this version of recognition of re-visited places is offered for the first time in the context of SLAM applications. The algorithm could be divided into two levels. Essentially, the task of all levels is consistently reducing the number of relevant images in accordance with the criteria defined for a particular step. The result of the algorithm is two matched images, the query image and its best-matched candidate that we will call a winner found in the database or lack thereof if the winner was not found.

The image retrieval process is preceded by a preparatory step. In this step, the k-means [27] clustering algorithm is run on a set of GIST descriptors of N independent images. This step produces a codebook $\{c_1, c_2, \cdots, c_n\}$ of n-centroids. Then the k-means algorithm is run on an independent set of SIFT descriptors to produce a codebook $\{s_1, s_2, \cdots, s_k\}$ on k-centroids. It is necessary to create the BOW model of an image representation. The BOW model represents every image as the histogram of appearance of a set of local features.

When the robot starts to move in the environment, it runs the SLAM algorithm and simultaneously creates an image database. The gist features of the new acquired image are assigned to the nearest cluster of the codebook $\{c_1, c_2, \cdots, c_n\}$. The criterion of similarity is the Euclidean distance between two vectors:

$$d_{min} = \| g - c_k \|, \qquad (2.8)$$

where g is the gist representation of a new image and k is the k^{th} centroid of the codebook of gist features $\{c_1, c_2, \cdots, c_n\}$.

Simultaneously, the robot extracts a set of local features and computes its BOW model. But not all images are added to the database. At the speed of the robot of 3 km per hour, encoded images are added to the image database with a frequency of 1Hz. The experimental results show that this frequency saves technical resources of a robot, it occupied an amount of memory, and allowed us to store sufficient amount of images for reliable place recognition. When the robot travels a certain distance and a sufficient amount of images will be collected in the database, the image retrieval algorithm is started.

The retrieval algorithm begins with a filtering step, which is based on the similarities of GIST descriptors (Figure 2.10). A gist feature of a query image is assigned to the nearest cluster of the codebook $\{c_1, c_2, \cdots, c_n\}$ according to equation (2.8). All of the images that are interior of the cluster are considered as relevant images or potentially matching images.

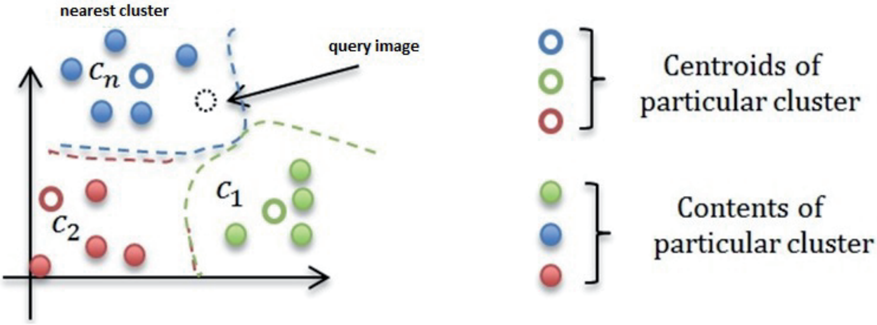

nearest cluster
query image
c_n
c_1
c_2
Centroids of particular cluster
Contents of particular cluster

Figure 2.10. Graphical representation of the first step. Filtering method based on the comparison of gist features. Here the query image is assigned to the nearest cluster $\boldsymbol{c_n}$*.The distance* $\boldsymbol{d}$ *between the query image and the centroid* $\boldsymbol{c_n}$ *is minimal.*

Practical experiments show that there is a need to consider m nearest clusters. The number m is chosen empirically and depends on many factors, like the context of the environment, the number of images in the database, the number of clusters in the codebook. Although the number of potential candidates was much smaller than the total number of images in the database, we reduced the number of relevant images by comparing of gist descriptors (Figure 2.11).

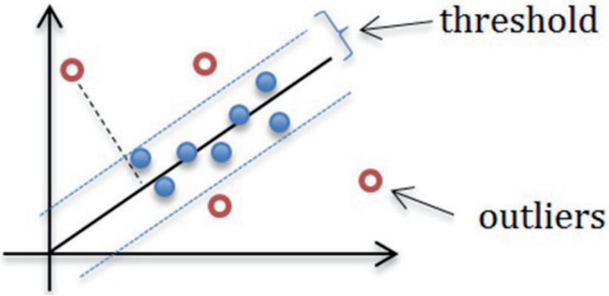


Figure 2.11. Verification of similarity of two GIST descriptors.

The GIST descriptor of query images was sequentially compared to all descriptions of potential candidates. At this stage, the threshold value was introduced. The Euclidean distance was computed and verified the condition of entering in the required range. All the images placed outside dotted lines (Figure 2.11) were rejected from the list of potentially matched images and not considered any more. The threshold value was also estimated by experience.

The result of the first step is the reduced set of relevant images. Typically, the first step filters out about 99% of database images.

In the second step, we measured the similarity between the distributions of local features in the two images. In order to know if the two images match, the distance between their BOW models was computed. As the image descriptor is a histogram, the Bhattacharyya's distance would be the best option [48]. The Bhattacharyya measure has a simple geometric interpretation as the cosine of the angle between the N-dimensional vectors: $(\sqrt{l(1)},\cdots,\sqrt{l(N)}\,)^T$ and$(\sqrt{l'(1)},\cdots,\sqrt{l'(N)}\,)^T$, where N is the length of the codebook $\{s_1,s_2,\cdots,s_k\}$.

Thus, if the two distributions are identical, we have:

$$\cos(\theta) = \sum_{i=1}^{N}\sqrt{l'(i)l(i)} = \sum_{i=1}^{N}\sqrt{l(i)l(i)} = \sum_{i=1}^{N} l(i) = 1,$$

and $\theta = 0$.

The metric distance between the two distributions could be estimated:

$$d(l,l') = \sqrt{1-\rho(l,l')}\,, \tag{2.9}$$

where $\rho(\cdot,\cdot)$ denotes the Bhattacharyya coefficient and could be estimated $\rho(l,l') = \sum_{i=1}^{N}\sqrt{l'(i)l(i)}$.

If the distance is under a predefined threshold, the images are said to match. The images potentially matching can be geometrically verified using the image position of the local feature. Assuming that the scene is rigid, the position x_i^1 of the feature i in image 1 and the position x_i^2 of the same feature i in image 2 are related by the epipolar constraint [18]:

$$x_i^{2^T} F x_i^1 = 0, \tag{2.10}$$

where F is the fundamental matrix that encodes the geometrical relation between the two views.

The minimum case for the estimation of the fundamental matrix needs seven point correspondences. The simplest method is the normalized 8-point algorithm [18]. After this step, only those potential matches that hold equation (2.10) are considered geometrically consistent and the rest are discarded.

The winner is an image that holds geometrical verification and has maximum positive matched points.

2.6 Experiment

In this part the image retrieved system performance is evaluated. Two high-quality multisensor image datasets from the RAWSEEDS project [57] extended with associated ground truth were used. To produce a codebook a set of images acquired by a front camera of a robot from BOVICA (outdoor + mixed) dataset was applied (algorithm 1). This algorithm is executed once at the preliminary stage in offline mode and requires a considerable amount of time for producing the codebook.

Algorithm 1. Producing a codebook of k*-clusters.*

```
Objective
    Given a set of N images, compute the codebook of $k$-clusters.
Algorithm
    compute the GIST  descriptor of  images
    run k-means algorithm to produce the codebook of $k$-centroids.
```

During the movement of the robot, each new incoming image is represented as a structure (2.7). To extract the GIST descriptor, the code proposed in [http://lear.inrialpes.fr/software] was used. To extract a set of local descriptors, the famous OpenCV library is the usual practice. As seen in Table 2.1, the algorithm for the extraction of a global descriptor from the image is much more efficient than the local features extractors.

Table 2.1. Image preprocessing time for extraction of set of SIFT and a GIST descriptors.

Local (SIFT) using OpenCV library	0.68 sec
Global (GIST) using the function coded in C	0.09 sec

Every new GIST descriptor is assigned to the nearest cluster (algorithm 2). In the experiment 52,700 images from the image sequence of “BICOCCA” (indoor) dataset were used.

Algorithm 2. Distribution of N gist descriptors.

```
Objective
    Given a set of N images from the image sequence and the codebook of $k$ −centroids, compute the image representation structure and store it. Assign each new GIST descriptor to the nearest cluster of the codebook.
Algorithm
    compute GIST descriptors  and a  set  of  SIFT descriptors, formula(2.7)
    assign all gist descriptors  to the  nearest cluster, formula (2.8).
```

The result of the algorithm is a database in which every image is a part of a particular cluster. This distribution of 52,700 images of the sequence acquired by the front camera mounted on the robot is shown in Figure 2.12.

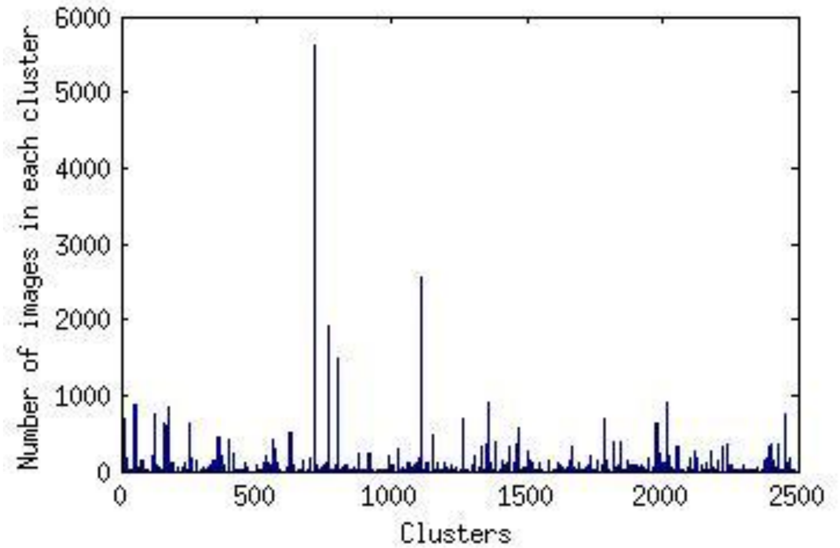


Figure 2.12. Distribution of 52,700 database images on clusters.

When the database is created, the robot may start to check every new image to recognize a place that it has seen before (algorithm 3).

Algorithm 3. Image retrieval algorithm.

Objective

Given a query image and a database of images. Retrieve the image from the image database that is similar to the query image.

Algorithm

1. **step 1 (global verification):**
   - assign a GIST descriptor of the query image to the m - nearest clusters of the codebook. Produce a list of relevant images;
   - compute the Euclidean distance between the gist of the query image and the gists from the list of relevant images;
   - from the list of relevant images reject all images whose Euclidean distance is greater than the threshold tr;
2. **step 2 (local verification):**
   - compute the distance between their bag-of-words model;
   - compute the number of matched local features on two images N (query and potential matched images);
   - using the image position of the local feature verify geometrically for all images with N > 8.

To simplify the analysis the algorithm is represented in two parts. Part 1 is designed to produce a list of images, which could be likely candidates for the query image. The number of images in the list should be considerably less than the total number of images stored in the database and depends on several factors, such as the size of the codebook, the degree of similarity between the scenes and the degree of similarity between GIST descriptors, respectively. As seen in Figure 2.12, each cluster contains a number of images. On average, this value is less than 900. However, four of the clusters involve many more images. One way to influence the number of images in the clusters is to change the size of the codebook. For example, Figure 2.13 shows the distribution of the same sequence of images on 1000 clusters. Based on this, the intuitive conclusion is: increasing the size of the codebook should reduce the number of GIST descriptors included in each cluster and hence leads to a reduction of the list of likely candidates. In this experiment, a list of less than 10 percent of database images was obtained. The next phase of the global filtering reduced this list to one percent after the estimation of the Euclidean distance between the GIST descriptor of the query image and the GIST descriptors of other images that belong to the list of likely candidates. At this point, the threshold value tr was introduced. The dependence of the number of images in the list of the threshold tr is shown in Table 2.2.

Table 2.2 The dependency of the number of images in the list of likely candidates of the threshold value.

Threshold	$tr_1 = 0.68$	$tr_2 = 0.75$	$tr_3 = 0.85$	$tr_4 = 1$
Number of images in the list	61	384	543	986
Percentage of images in the list	0.11%	0.73%	1.04%	1.89%

When the value of the threshold $tr = 0.68$, the average number of images in the list of likely candidates is 61, which corresponds to 0.11 percent of the total number of images. Increasing the threshold tr leads to increasing the list of likely candidates. To understand the reason of introducing the threshold the following discussion is helpful. “M” nearest clusters that were successfully identified in the first step of algorithm 3, contains a number of images. Some of them perhaps are desired candidates, but most of the list should be rejected. The threshold tr reduced the list of likely candidates and all images that were outside the circle with a radius tr (Figure 2.14, right image) had to be excluded.

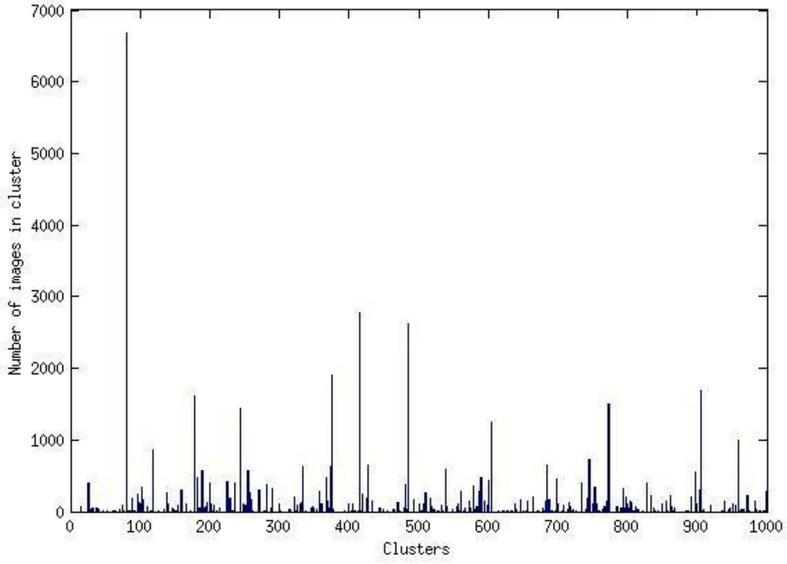


Figure 2.13. Distribution of 52700 database images on 1000 clusters

.

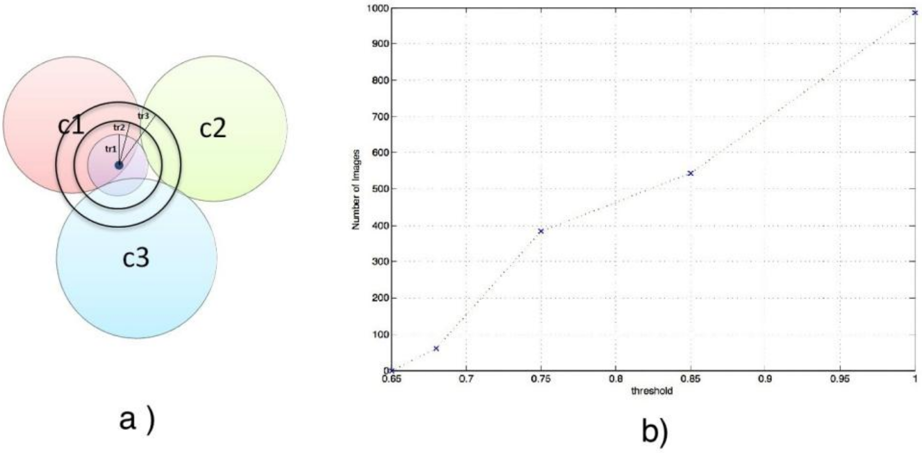


Figure 2.14. Dependence of the number of images in the list of likely candidates on the threshold value: a) three independent nearest clusters with centroids $c1$, $c2$, $c3$. The query image is included in the red cluster with the centroid c1. Three different circles with the center in the query image and radius $\boldsymbol{tr}$ define a list of likely candidates, which here is $\boldsymbol{tr1} < \boldsymbol{tr2} < \boldsymbol{tr3}$. The graph on the left image (b) shows how the size of the list changes if the threshold value is changed.

Theoretically, it is difficult to estimate the magnitude of the threshold. Practical experiments with dataset images from the RAWSEED project [57] showed that for a threshold value less than $tr1$ ($tr1 < 0.68$) the list remains empty. It means that the Euclidean distance between the images is greater than the threshold value $tr1$ and all images from the database are outside the smallest circle with the radius $tr1$ (Figure 2.14 (a)). If we increase the threshold, the first nearest images will be included to the list. The slope of the graph (Figure 16 image b)) on the segment from $tr1$ to $tr2$ is equal to $m = \frac{nImages_2 - nImages_1}{tr_2 - tr_1} = \frac{384-61}{0.75-0.68} = 4.6 \cdot 10^3$. The graph on the segment from $tr2$ to $tr3$ is more sloping $m = \frac{nImages_2 - nImages_1}{tr_3 - tr_2} = \frac{543-384}{0.85-0.75} = 1.59 \cdot 10^3$. This means that the size of the list grows slowly. Intuitively, the desired candidates should be closer to the query image than other images. When the majority of the nearest images are included, we change the threshold value until $tr3 = 0.85$ to increase the probability that the desired candidates are in the list. As can be seen from Figure 16 a), the desired candidates may be part of different clusters. In fact, increasing the value of the threshold expands the circle around the query image and adds to the list of likely candidates not only desired candidates but any other images. The list becomes redundant. The analysis of the graph has showed that the increase of the threshold tr beyond the saturation threshold (in this case $tr_3 > 0.85$) is not justified and leads to an increase of the list of possible candidates, and to growing the retrieval time, respectively. Thus the list of likely candidates is reduced (Table 2.2) and contains no more than one percent of images.

Part 2 of algorithm 3 is designed to further reduce the number of likely candidates and determine a winner. The images potentially matching can be geometrically verified using the image position of the local feature. In order to know if two images match, the distance between their BOW models is computed. A popular method described in [27] should filter out all undesired images from the list. Without this step, all of the images from the list should be checked for local points matching. For all images with $N > 8$, geometrical verification using the image position of the local feature is carried out, where N is the number of corresponding points. An image containing a large amount of corresponding points that have successfully passed the geometry test is chosen as a winner. It is important to emphasize the need for geometric verification. Figure 2.15 shows the common situation the robot faces by operating in the indoor environment. The images look very similar. Even the human eye cannot distinguish that both scenes were captured in different places. Lack of geometrical checks could lead to false positives of the system.

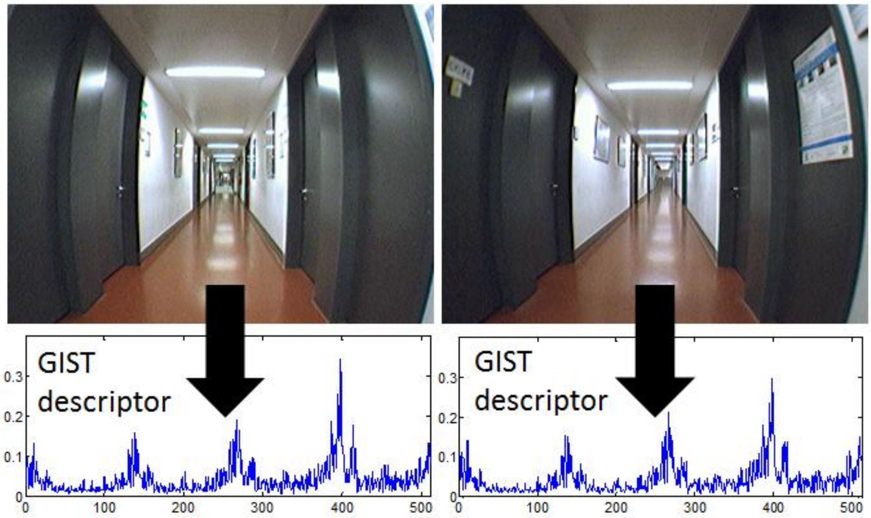


Figure 2.15. Absolutely different places but they look very similar. Moreover, the GIST representation of both images is also very similar.

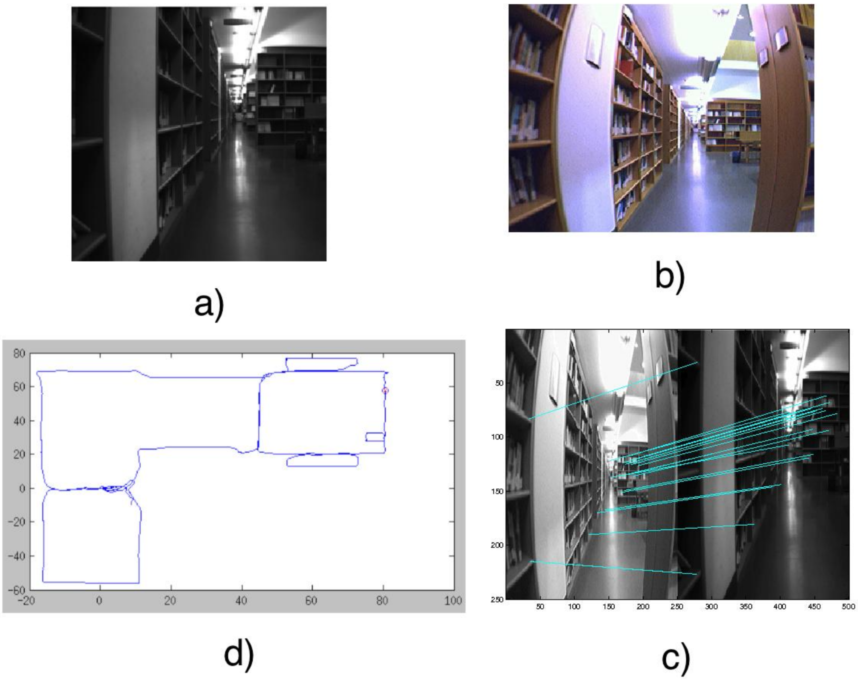


Figure 2.16. Result of the retrieval algorithm: a) query image, b) winner, c) a pair of matched images with a set of corresponding points, d) ground truth path.

The result of the retrieval algorithm (algorithm 3) is shown in Figure 2.16. The blue circle on the path marks the position of the robot by taking the query picture and the red cross is the position of the robot by taking the winner pictures.

Each image included to the RAWSEED datasets contains information about the true position of the robot. This allows us to make a visual check. The part of the robot path in an enlarged scale is shown in Figure 2.17. A set of corresponding points in both images (query and winner images) is shown in Figure 2.18.

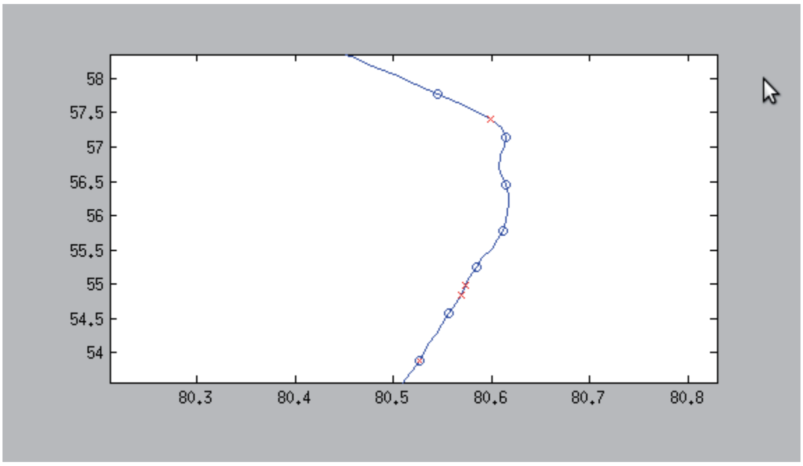


Figure 2.17. A part of the map. Blue circles are the positions of the robot by shooting a query image and red crosses mark the positions of the robot by shooting a winner image.

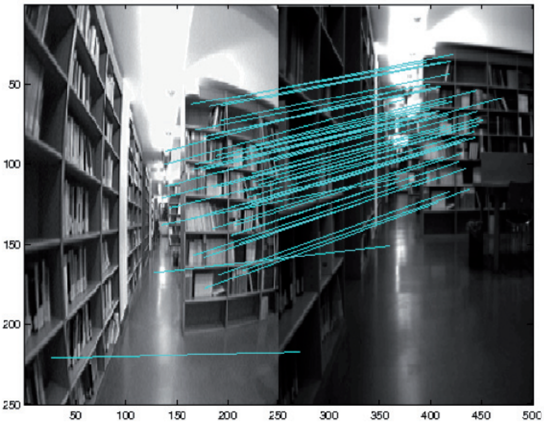


Figure 2.18. A set of corresponding local features on the query and winner images.

Usually, the properties of the retrieval system are evaluated using a precision vs. a recall graph. The graph shows how the accuracy of the system is reduced while increasing the fraction of retrieved images. The precision and the recall could be defined with the equations:

$$\text{Precision} = \frac{r}{N}, \qquad (2.11)$$

$$\text{Recall} = \frac{r}{R}, \qquad (2.12)$$

where r is the number of relevant images retrieved, N is the total number of images retrieved; R is the total number of relevant images in the database.

The number of retrieved images is regulated by the threshold value tr. The results of the evaluation of the efficiency of the image retrieval system based on the global descriptor for different threshold values tr are presented in Table 2.3 and on the graph (Figure 2.19). Five additional images of the outdoor environment have been included in the database. Different types of scenes represented in the database and added to the database are used for visual clarity, and do not affect the results of the experiment.

Table 2.3 The properties of the retrieval system.

Threshold	Total number of images retrieved N	Precision $\frac{r}{N}$	Recall $\frac{r}{R}$
0.2	1	1	1/5
0.23	2	1	2/5
0.25	3	1	3/5
0.27	4	1	4/5
0.3	5	5/5 =1	1
0.68	31	5/31 = 0.16	1
0.8	501	5/501 = 0.01	1

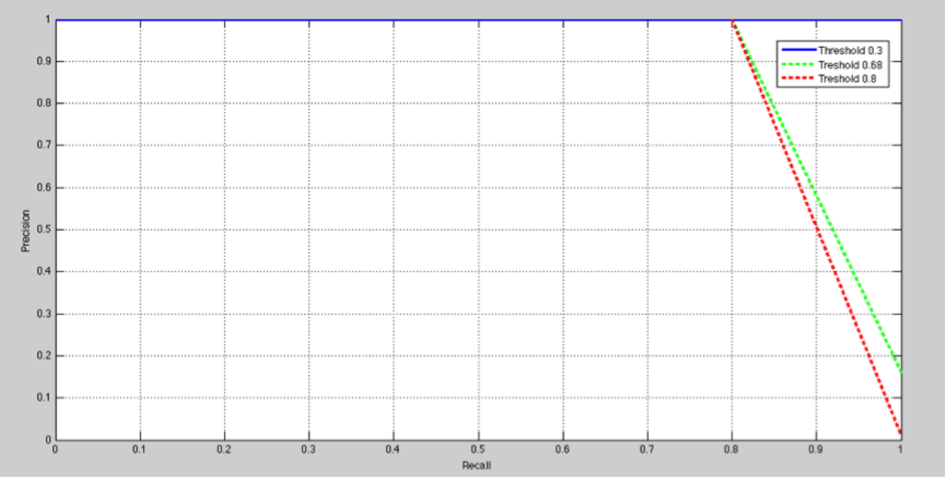


Figure 2.19. Precision vs. recall graph for different thresholds.

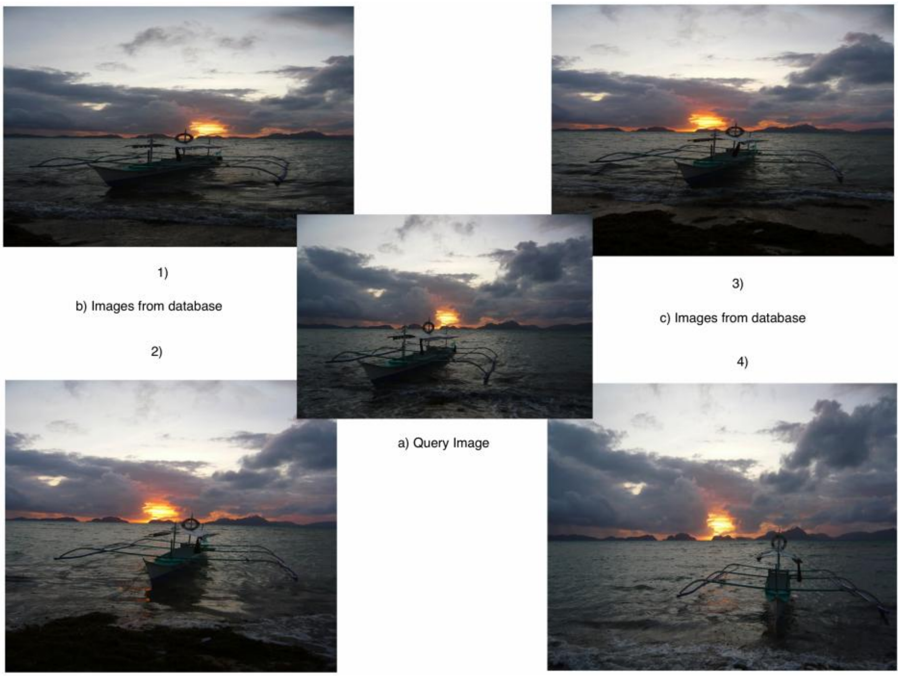


Figure 2.20. A set of images was used to evaluate the properties of the image retrieval system based on the global scene representation: a) the query image 1-4) images added to the database.

The experiments showed that the accuracy of the image retrieval system based on the global properties of the scene decreases rapidly with the increase of the threshold. Experiments with real “BICOCCA” data [57] showed that the average threshold value is equal to 0.8, where for all 200-query images the same number of matched images was found. The threshold magnitude less than 0.8

leads to the conclusion that a desired candidate will not be found for all query images. The magnitude higher than $tr = 0.8$ makes the list of the likely candidates redundant.

2.7 Conclusion

As it was shown at the beginning of this chapter, different ways to solve the map-merging problem can be borrowed from the robots capable of figuring out the loop-closing problem. The essence of both of the problems is the same: to find an image in the database, which is similar to the query image. As is known, the solving of the loop-closing problem is far from perfect. This chapter proposes a new efficient method. It operates successfully with huge amounts of images. This is the main advantage of the method over the previously existing methods. The multi-stage filtration procedure based on the comparison of global properties of two different scenes has made the method successful. To increase the accuracy of the image retrieval system the local points matching procedure and finally the epipolare geometry constraints verification is incorporated. After series of successful experiments with the data obtained from one robot, it becomes possible to extend this image finding strategy to a multi-robot case. In such situations, the new database will be much larger and will save the information from several robots. The image retrieval system will operate more efficiently with a large image database if several additional requirements are met:

- The dimension of the codebook should be universal. It is created in the offline mode and should pass to any such system. To achieve it, the number of clusters or the dimension of the codebook has to be much larger. In [11] the author used the codebook with k = 20 000. This allows us to find a similar image in a huge database with more than 1000 000 images.
- By construction of the universal codebook an independent set of images with the widest possible range of different types of scenes has to be used.
- Adhering to these requirements will prevent cases like accumulation of a huge number of images in one cluster, but a few others at the same time remain empty.

With these results we can return to the multi-robot case and to the map-merging problem. If a robot gets from the system an image, which is similar to the query image, the task of combining two local maps is reduced to the transformation of 3D points of one coordinate system to another. The statement of the map-merging problem and possible solutions are the subject of the next chapter.

3. Map-merging problem in a multi-robot application

3.1 Introduction

Chapter 2 discussed the characteristics of the methods for alignment of two local maps in a global map. The landmark-based method has significant advantages over other methods and offers a basic strategy to address this problem. This strategy involves three steps:

- Let given two sets of 3D points Q and M have an overlapping area. Find a set of corresponding point-pairs.
- Given a set of corresponding point-pairs, a rigid transform can be computed that best positions the two shapes so that the distance between the corresponding points is minimized.
- Decide if a given refined estimate is correct.

To continue the study of the alignment process, it is necessary to clarify what the 3D map created by a mobile robot is. In this thesis, the local map is defined as a set of vertices in a $\mathbb{R}^3$. Each vertex is defined by 3D coordinates (x, y, z) and represents the outer surface of the object. In the computer literature, such a set of points is called a point cloud. Generally, to build a 3D point cloud, a 3D scanner is commonly used. Recent visual SLAM equipment has all advantages of video cameras and provides in-depth information of the scene. Such devices belong to the group of active vision sensors and consist of a color camera and a sensor of depth. An example of such a device is a KINECT camera [1], [25], [3]. The depth sensor consists of an infrared projector combined with a monochrome camera, which allows the sensor to obtain three-dimensional KINECT-images of a scene. Due to high technical characteristics combined with the low cost, the KINECT camera is widely used in SLAM applications.

Due to the intensive use of high-accuracy measurement tools for sensing of a 3D space, in the robotics community, the affine registration problem is intensively discussed. The registration problem is defined as a problem of estimation of the unknown parameters of the transformations between two 3D scans and finally, the aligning of these scans. Obviously, the map-merging problem and the registration problem are very similar and could be considered as identical. At present various algorithms are available for solving the registration problem. Under certain restrictions, they successfully solve this task. However, all of these algorithms have drawbacks. In parallel, research in the field of neural networks is gaining momentum. Neural network algorithms successfully solve the pattern recognition problem, data classification, and the nonlinear approximation problem. In addition, artificial neural network has a number of advantages over traditional computation methods. Powerful computing potential of the neural network is able to solve a map-merging problem as effectively as the traditional methods.

This chapter focuses on the potential of artificial neural network for solving the map-merging problem.

3.2 Review of existing methods

Modern 3D high-accuracy measurement tools have simplified the scanning process of the environment. Without a priori knowledge of the relative displacements of the active sensors, the registration makes use of a registration of partially overlapped images. A typical formulation of the registration problem could be described in the following way: two sets of 3D points or a 3D point cloud, commonly called a model and a date, are given. The aim is to find transformation parameters that optimally merge two 3D point clouds. The affine transformation in $\mathbb{R}^3$is given by matrix A. For any point $d_1(x_1, y_1, z_1) \in D$ the matched point $m_1(x_2, y_2, z_2) \in M$ could be found by the equation: ${m_1}^T = A \cdot d_1^T$. In other words, the solution of the problem is reduced to finding the nine variables by solving a system of nine linear equations. Because a 3D point has three degrees of freedom, it is necessary to specify three-point correspondences in order to constrain A fully. If exact three-point correspondences are given, then the exact solution is possible. But in practical applications, the coordinates of points are measured inaccurately. If more than three-point correspondences are given, then one attempts to estimate an approximate solution, namely transformation parameters that minimize some cost function.

Existing methods for solving the registration problem could be divided into two classes:

- class of voting methods;
- class of methods using a corresponding point pair.

Geometrician Hashing [63] refers to the first class of methods. Initially, the method was developed for computer vision applications and designed for pattern recognition. The recognition system was able to recognize objects on the image presented partly or undergoes geometrical transformations. In such a system, models of objects are stored in a database. To quickly find a model object in the database, a method of sequential search does not apply. The method provides direct access only to specific information that would give the best result. These classes of methods are particularly attractive for model-based schemes, but also for object comparison. The algorithm consists of two phases: a preparatory phase and a search phase. To understand the advantages and disadvantages of this class of methods, below the basic idea is explained in more detail.

The preparatory phase takes place in an offline process. The system encodes data about the model and stores it in a database, in this case called the hash table. In the recognition phase, indexing-geometric properties of the features of a scene are extracted and matched with candidate models, which are stored in a hash table. Each match increases the number of votes in favor of the model. If the model has acquired a sufficient number of votes, one can expect that the table cells contain potential candidates that determine the object similarity. For each potential match, the transformation matrix is estimated. The transformation that has the highest number of successful mappings for a given error value is selected as the optimal transformation.

It should be noted that geometrical hashing is used in both 2D as well as in a high-dimensional case. The method is widely used for pattern matching in computer vision, CAD/CAM, medical image processing, molecular biology, and medicinal chemistry. However, since methods of this class tend to be costly, they are not commonly used for global registration of scan data. The time complexity of the recognition phase is $O(HS^{c+1})$, where H represents the complexity of the hash table bin, S is the set of scene features and c features are needed to form a basis.

The second class of methods makes use of the point correspondences from two 3D point clouds. If the correspondence $\{d_i \leftrightarrow m_i\}$ exists, the rigid transform can be computed so that the distance between the corresponding points is minimized. In order to minimize the distance between two point clouds the iterative closest point (ICP) [2] algorithm or its variants are commonly used. The algorithm assigns to each point from one cloud the closest point from the second cloud, and then the transformation could be estimated. The process is iterated until some convergence criterion is reached. Mathematically, given 3D-to-3D points correspondences $\{d_i \leftrightarrow m_i\}$ determine the 3D rigid transformation for which the error is minimized:

$$\min E(R,t) = \frac{1}{N_p}\sum_{i=1}^{N_p}\|m_i - Rd_i - t\|^2. \tag{3.1}$$

This idea is plotted in Figure 3.1.

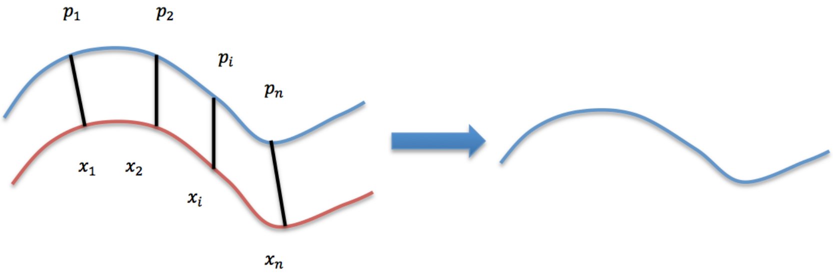


Figure 3.1. 2D transformation problem.

The transformation problem can be solved as follows: first, the center mass $\mu_x = \frac{1}{N_x}\sum_{i=1}^{N_x} x_i$ and $\mu_p = \frac{1}{N_p}\sum_{i=1}^{N_p} p_i$ are calculated. From every point of the point cloud, the corresponding center mass is extracted. The resulting point sets are: $X' = \{x_i - \mu_x\} = \{x_i'\}$ and $P' = \{p_i - \mu_p\} = \{p_i'\}$. The transformation parameters could be calculated from matrix $W = \sum_{i=1}^{N_p} x_i' p_i'^T$ by the SVD decomposition:

$$W = U\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & 0\\ 0 & 0 & \sigma_3\end{bmatrix}V^T. \tag{3.2}$$

Here U, V are 3 by 3 matrixes and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are the singular values of W. For matrix W the optimal solution of $E(R, t)$ is unique and is given by $R = UV^T$ and $t = \mu_x - R\mu_p$. The minimal value of the error function $E(R, t)$ is

$$\min E(R, t) = \sum_{i=1}^{N_p} \|x_i'\|^2 + \|p_i'\|^2 - 2(\sigma_1 + \sigma_2 + \sigma_3). \qquad (3.3)$$

The ICP algorithm is often used to reconstruct 2D or 3D surfaces from different scans. The main disadvantage of this method is that it does not guarantee finding the globally optimal alignment, and therefore is only effective when the initial position of the input shapes is close to the correct alignment [14]. Probably, the ICP algorithm could be successfully used for maps alignment, but existing shortcomings activate to further search for optimal solutions.

3.3 Artificial Neural Network

Artificial neural network, often called a neural network (NN), is a mathematical model of biological neural networks. An example of biological neural networks is the human brain. The human brain is an extremely complex, nonlinear parallel computer. The brain has the ability to organize its structural components so that they could perform specific tasks many times faster than any modern computer. An example of such a problem can be a problem of identification of a familiar face from a number of unfamiliar faces. The human brain performs this task for 100-200 milliseconds [19]. The computer takes several days to perform the same task. The basic element of a neural network is a neuron. A neural network consists of a set of neurons connected in a certain way. During the training, the NN accumulates the experimental knowledge and uses it for further processing. The procedure used for the learning process is called a learning algorithm. Training algorithms adjust the synaptic weights in a certain order to provide the necessary interconnection structure of neurons. The power of NNs lies in the parallelism of computational processing and in the self-learning ability. Such properties of the NN as nonlinearity, input-output mapping, adaptivity, fault tolerance, uniformity of analysis and design provide significant advantages over conventional information processing systems. Today, the NN, a quick and powerful computational tool, has been successfully applied in practice, like in classification problems, pattern recognition, filtering, signal processing.

3.3.1 Model of an artificial neuron

An artificial neuron is a nonlinear elementary computational element of a neural network that consists of one or more inputs, a weighted adder and an output. A model of an artificial neuron is shown in Figure 3.2.

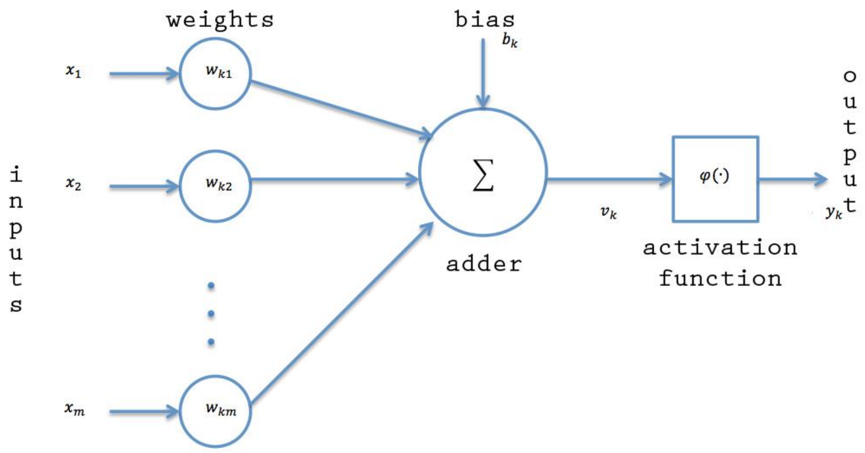


Figure 3.2. Nonlinear model of an artificial neuron. [19]

A mathematical model of a neuron is given by the following expressions:

$$y_k = \varphi(u_k + b_k), \tag{3.4}$$

$$u_k = \sum_{j=1}^{m} w_{kj} x_j, \tag{3.5}$$

where x_1, x_1,⋯ x_m– neuron's input nodes, w_{k1}, w_{k2},⋯ w_{km} - synaptic weights of a neuron, u_k - linear combiner output, $\varphi(\cdot)$ - activation function, y_k – output signal of a neuron k . The only output of the neuron k may take a digital or an analog value.

The presented model of an artificial neuron known as the McCulloch-Pitts model was first presented in [42].

3.3.2 Neural network architectures

The architecture of the neural network is closely related to learning algorithms, and these in turn depend on the task, the solution of which must the NN be trained to achieve. In general, there are three main classes of network architectures: single-layer feedforward networks, multilayer feedforward networks, and recurrent networks. In NNs neurons are arranged in layers. Single-layer networks are the simplest case of a multilayer network. Single-layer network is composed of input layer neurons and output layer neurons. Information flows in one direction from the input layer to the output. Multilayer network is characterized by the presence of one or more hidden layers, which are intended to highlight the global properties of the data using the local connections between the neurons. The recurrent network has at least one feedback loop.

3.3.3 Approximation of nonlinear functions

One of the tasks to be solved by a multilayer network is the problem of approximation of a nonlinear function. A mapping of input data set into outputs is given by

$$d = f(x), \tag{3.6}$$

where x - the input vector and d - the output vector.

Function $f(\cdot)$ is the function that should be found. In order to configure the network for the implementation of the transformation, a set of training data is required:

$$T = \{(x_i\,, d_i)\}_{i=1}^{N}\,, \tag{3.7}$$

Requirement for the neural network that approximates the unknown function $f(\cdot)$: function $F\ (\cdot)$, which describes the mapping of the input signal to the output, should be close enough to the function $f\ (\cdot)$ in terms of the Euclidean norm on the set of all input vectors x:

$$\|F(x) - f(x)\| < \varepsilon, \tag{3.8}$$

for all vectors x, where ε is a small positive number.

The process of configuration of synaptic weights is called a learning process. To illustrate the learning process, a classical error-correction learning algorithm for the neuron k is considered. The neuron k receives an input vector $x(n)$, where n is the discrete time or the number of the step of the iterative learning process. Output of the neuron k is denoted by $y_k(n)$. During training, this output will be compared with the desired response $d_k\ (n)$. As a result; the error signal $e_k(n)$ is given by the equation

$$e_k(n) = y_k(n) - \ d_k(n). \tag{3.9}$$

The error signal triggers the correction of synaptic weights of a neuron k. These corrections should bring the output signal of the neuron k to the desired state and could be achieved through the minimization of the cost function. The function $E\ (n)$ determines the instantaneous value of the error energy and is given by

$$E(n) = \frac{1}{2} e_k^2(n). \tag{3.10}$$

Minimization of the cost function is performed by the delta rule [19]. Let $w_{kj}(n)$ be the current value of the synoptic weight w_{kj} of neuron k, corresponding to an element $x_j(n)$ of the vector $x(n)$ on the sampling step n. The updated weights of the neuron are

$$\Delta w_{kj}(n) = \ \eta e_k(n) x_j(n), \tag{3.11}$$

where η is a positive constant, which determines the learning speed.

The new value of the synaptic weights for the next step is given by

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n). \tag{3.12}$$

3.3.4 Multilayer perceptron

Multilayer perceptron, an extended version of a single-layer perceptron, consists of multiple sensor elements forming the input layer, one or more hidden layers and an output layer. The input signal is transmitted in one direction from layer to layer. The popular error back-propagation algorithm is a standard learning algorithm used for training of a multilayer perceptron [19]. This algorithm is based on an error correction learning rule. Multilayer perceptron differs from other architectures by the following features:

- Every neuron has a nonlinear activation function. A nonlinear activation function allows approximating the nonlinear "input-output" mapping. Sigmoidal nonlinear function is one of the popular activation functions.

$$y_j = \frac{1}{1+\exp(-v_j)}, \tag{3.13}$$

where v_j - induced local field of neuron j and y_j - output of neuron j.

- The network has one or more hidden layers. Neurons from these layers extract the most important features from the input data.

A full-connected network with one hidden layer is shown in Figure 3.3.

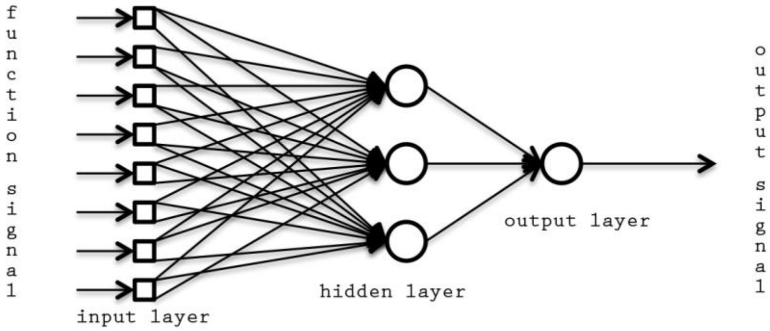


Figure 3.3. The graph of a multilayer perceptron with a hidden layer.

As already mentioned above, the multilayer perceptron is a powerful computing structure that is capable of performing a wide range of tasks. The approximation of a nonlinear function is one of such problems.

3.4 Map-merging problem as a task of nonlinear function approximation

In general, the map-merging problem can be considered as a registration problem. In this section the multilayer artificial neural network is used as a computational mechanism. First, we formulate the problem: given 3D data sets of a shape are captured from different viewpoints (Figure 3.4) [41].

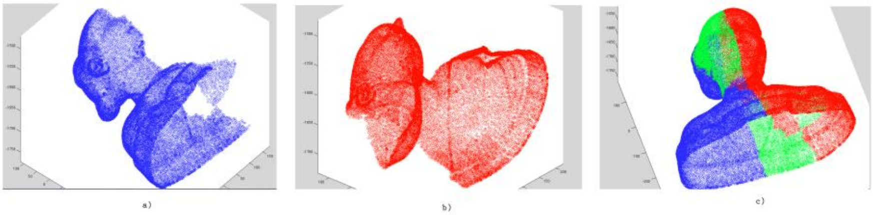


Figure 3.4. Range image of a model. (a) 3D data set of a mesh in the first sensor coordinate system, (b) 3D data set of the same mesh in the coordinate system of a second sensor, (c) integrated model in the common coordinate system.

The main goal is aligning multiple 3D data sets in a common coordinate system (Figure 3.4. (c)). No prior information about a sensor's position is known. We assume only that two views contain overlapping scene regions (overlaps). These regions are marked in green in Figure 3.4 (c). The ability of the neural network to approximate the unknown mapping of input space into the output is well known and used in identification system applications. The learning algorithm updates the synaptic weights so that the network will be able to approximate an unknown function $f\,(\cdot)$ in accordance with the expression (3.8). Block diagram of a possible solution of this task is plotted in Figure 3.5.

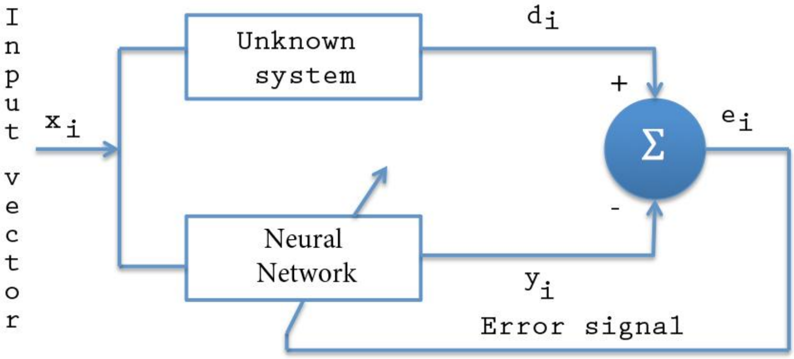


Figure 3.5. Approximation of an unknown function by an artificial neural network.

The output of the neural network $y_i = F\,(x_i)$ is compared with the desired response d_i. The difference between the network output y_i and the desired response d_i generates an error signal e_i. The error signal updates the free parameters of the network in order to minimize the mean square error. Such schemes can be used for the estimation of unknown transformation parameters existing between two scans (Figure 3.4). In this case, the unknown system represents any transformation matrix. It is worth remembering that the algorithm presented in the third part provides data about an existing overlap in two views. To move from overlapped image planes to overlapped 3D point clouds could be performed by the following technique. The set of 2D matched points on two views allowed us to estimate a 3 by 3 homography matrix. For example, for two images from Figure 3.6, the homography matrix could be estimated using an algorithm presented in [18] (direct linear transformation, RANSAC and Levenberg-Marquardt optimization):

$$H = \begin{bmatrix} 1.0785 & -0.0479 & -339.9057 \\ 0.0607 & 1.0286 & 6.1660 \\ 7.657e-05 & -3.388e-05 & 1 \end{bmatrix}.$$

Finally, a high accuracy measuring system (like KINECT) gives the correspondences for any 2D matching point-pairs on the image plane to the corresponding 3D matching point-pairs in the sensor's coordinate frame. In this case, the task of alignment of two local maps is reduced to the registration problem of 3D surfaces by known matching point-pairs. The set of corresponding 3D point-pairs from existing overlaps (marked with green in Figure 3.4) is used as training examples $T = \{(x_i, d_i)\}_{i=1}^{N}$, where N is the number of matched points.

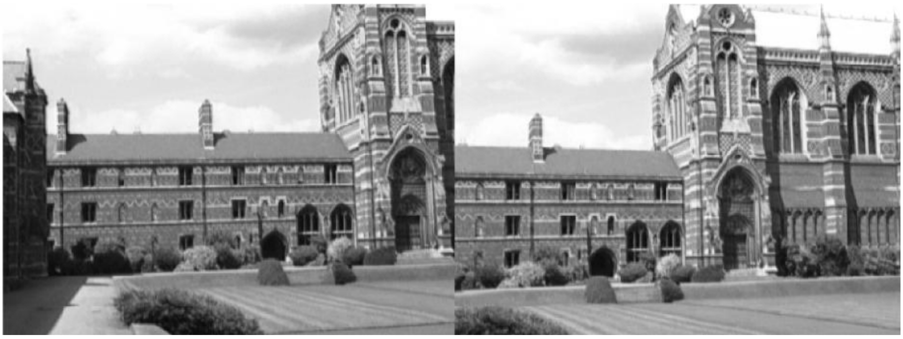


Figure 3.6. Two images created by a rotating camera [64].

The architecture of the neural network performs the identification task, as shown in Figure 3.7. The nonlinear activation function in a neuron provides an approximation of the nonlinear function.

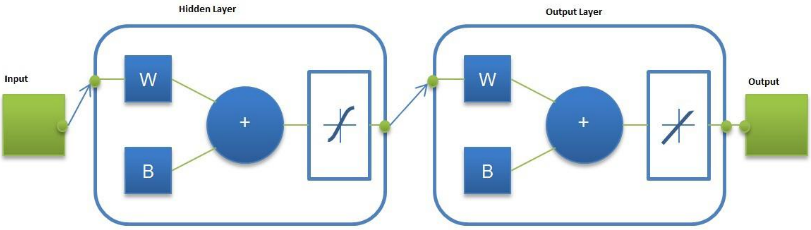


Figure 3.7. Multilayer perceptron for a function approximation.

Standard learning algorithm for multilayer feedforward networks is an error back-propagation algorithm [19]. Different research communities in different contexts have used this numerical method. This method is based on the calculation of partial derivation of the network function $F_i(x)$ with respect to the input x. For a neural network from Figure 3.7 each weight should be updated using the increment $\Delta w_{ji}(n)$:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n), \tag{3.14}$$

where $y_i(n)$ is an output signal of a neuron j, η is a learning constant and $\delta_j(n)$ is a local gradient.

If the neuron j is an output neuron, then the local gradient $\delta_j(n) = \varphi_j'(v_j(n) \cdot e_j(n)$, where $\varphi_j'(v_j(n))$ is a derivative of the activation function and $e_j(n)$ is an error signal. For the neuron j from hidden layers, the local gradient $\delta_j(n)$ is

$$\delta_j(n) = \varphi_j'\left(v_j(n)\right) \sum_k \sigma_k\,(n) w_{kj}(n), \tag{3.15}$$

where $\sigma_k(n)$ depends on the error signals $e_k(n)$ for all neurons of the output layer and $w_{kj}(n)$ is synaptic weights between the neuron k and j.

The popular activation function for a multilayer network $\varphi(\cdot)$ is a sigmoid (equation $y_j = \frac{1}{1+\exp\,(-v_j)}$), but a nonsymmetrical function has more advantages. The multilayer perceptron network can learn faster if the activation function is a hyperbolic tangent:

$$\varphi(v) = a \tanh(dv), \tag{3.16}$$

where $(a, b) > 0$. The best value for $a = 1.7159$ and $b = 0.666$ taken from [18] (Figure 3.9).

The derivate of the hyperbolic tangent function could by calculated by the following equation:

$$\varphi_j'\left(v_j(n)\right) = ab\,\mathrm{sech}^2\left(bv_j(n)\right) = ab\left(1 - \tanh^2\left(bv_j(n)\right)\right), \tag{3.17}$$

$$\varphi_j'\left(v_j(n)\right) = \frac{b}{a}\left[a - y_j(n)\right]\left[a + y_j(n)\right]. \tag{3.18}$$

Now we can write the backpropagation in the matrix form for a network with a hidden and an output layer. The derivatives for m output units stored in matrix $D^2$are

$$D^2 = \begin{pmatrix} \frac{b}{a}\left[a - y_1^{(2)}(n)\right]\left[a + y_1^{(2)}(n)\right] & 0 & \cdots & 0 \\ 0 & \frac{b}{a}\left[a - y_2^{(2)}(n)\right]\left[a + y_2^{(2)}(n)\right] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{b}{a}\left[a - y_m^{(2)}(n)\right]\left[a + y_m^{(2)}(n)\right] \end{pmatrix}. \tag{3.19}$$

The derivatives for k-hidden units stored in matrix D^1 are

$$D^1 = \begin{pmatrix} \frac{b}{a}\left[a - y_1^{(1)}(n)\right]\left[a + y_1^{(1)}(n)\right] & 0 & \dots & 0 \\ 0 & \frac{b}{a}\left[a - y_2^{(1)}(n)\right]\left[a + y_2^{(1)}(n)\right] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{b}{a}\left[a - y_k^{(1)}(n)\right]\left[a + y_k^{(1)}(n)\right] \end{pmatrix}. \quad (3.20)$$

Define the error vector e:

$$e = \begin{bmatrix} y_1^{(2)}(n) - d_1(n) \\ y_2^{(2)}(n) - d_2(n) \\ \vdots \\ y_m^{(2)}(n) - d_m(n) \end{bmatrix}. \quad (3.21)$$

The m-dimensional vector of the local gradient $\delta^{(2)}$ of the backpropagated error up to the output units is given by the expression:

$$\delta^{(2)} = D^2 \cdot e. \quad (3.22)$$

The k-dimensional vector of the local gradient $\delta^{(1)}$ of the backpropagated error for the hidden layer is given by the expression:

$$\delta^{(1)} = D^1 W_2 \, \delta^{(2)}, \quad (3.23)$$

where W_2 is a $(m \times k)$ matrix of weight:

$$W_2 = \begin{bmatrix} w_{11}^{(2)}(n) & w_{12}^{(2)}(n) & \cdots & w_{1k}^{(2)}(n) \\ w_{21}^{(2)}(n) & w_{22}^{(2)}(n) & \cdots & w_{2k}^{(2)}(n) \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1}^{(2)}(n) & w_{m2}^{(2)}(n) & \cdots & w_{mk}^{(2)}(n) \end{bmatrix}. \quad (3.24)$$

Finally, the corrections for matrices W_1 and W_2 are then given by

$$\Delta W_2^T = \eta \delta^{(2)} y(n), \quad (3.25)$$

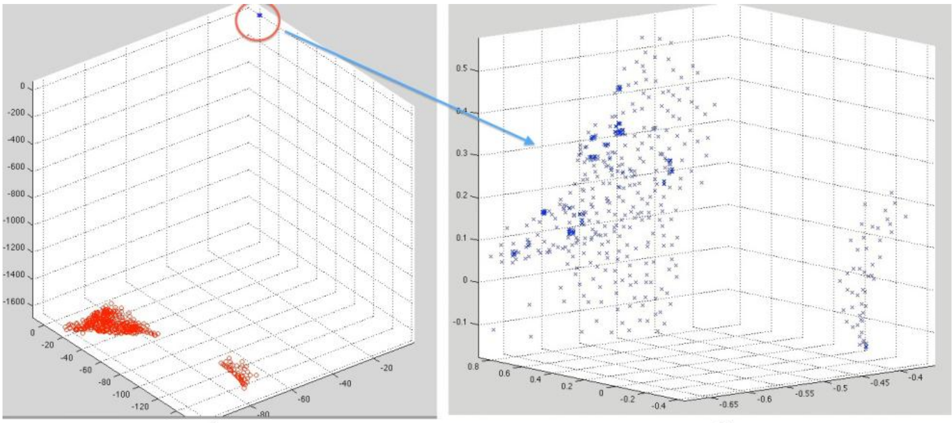
where $y(n) = \left[y_1^{(1)}(n) \quad y_2^{(1)}(n) \quad \cdots \quad y_k^{(1)}(n)\right]^T$ is a k-vector of the input signal for neurons of the output layer and

$$\Delta W_1^T = \eta \delta^{(1)} x(n), \quad (3.26)$$

where $x(n)$ is a vector of the input signal.

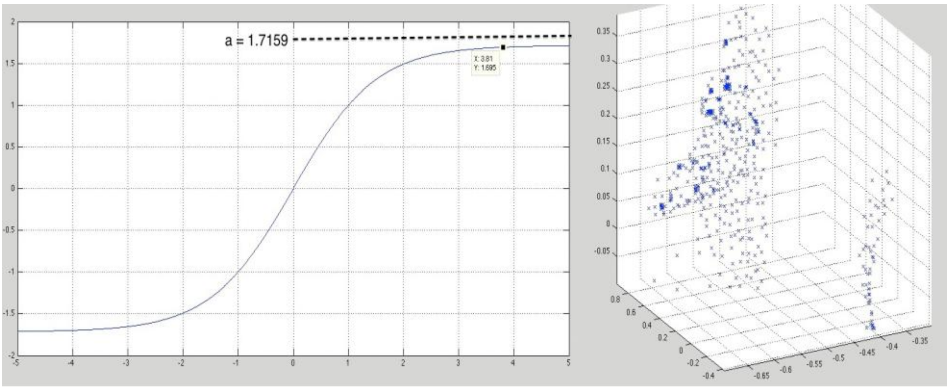
The new synaptic weights for the next step are given by equation (3.12).

Next, normalizing the inputs and target values [33] needs explanation. All input and data variables should be pre-processed, so that the mean value of all data from the training set should be close to zero, in order to compare inputs with standard deviation, and examples from the training sets belong to the linear interval of an activation function. Figure 3.8 shows an input data set plotted with red and a set of normalized inputs with blue.



a) b)

Figure 3.8. The set of 3D input points (a) and the normalized 3D coordinates (b) created by the three-step method (displacement of average value, decorrelation, and alignment of covariance).



a) b)

Figure 3.9. Hyperbolic tangent activation function a = 1.7159, b = 2/3 (a) and normalized set of training examples (b).

In general, the design of the neural network is a complex process in which multiple parameters of the process are determined by personal experience. In order to achieve higher speed training and the accuracy of approximation the existing heuristic recommendations should be taken into account. The next section is devoted to the discussion of the results of approximation of the unknown function with the neural network.

3.5 Result of the experiment

In this section, the experimental result of 3D affine registration by the neural network is discussed. The aim was to provide convincing empirical validation of the accuracy as well as robustness of the ability of the NN to approximate unknown transformation. To examine the NN a simplified version of 3D scanned mesh of Ippolita Sforza's statue sculptures by Francesco Laurana (Figure 3.10) was used.

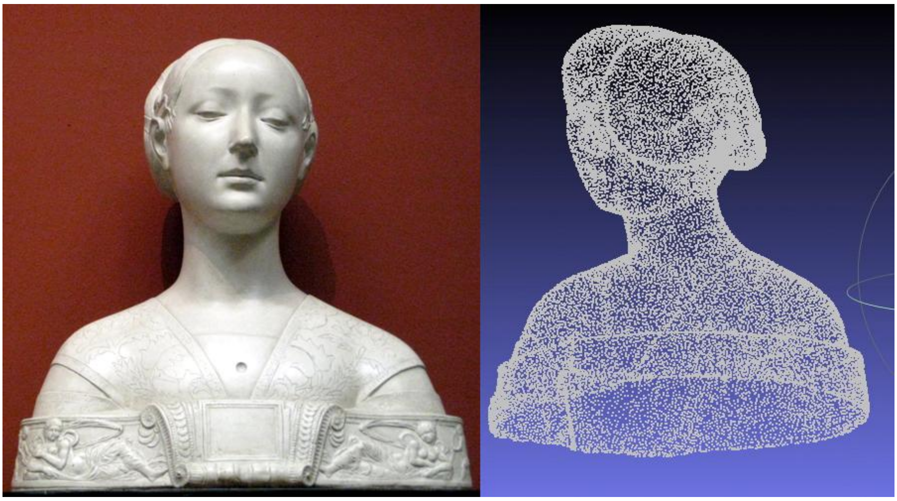


Figure 3.10. Francesco Laurana female bust (left), 3D scanned mesh (right) created by using MeshLab [41], a tool developed with the support of the 3D-CoForm project [65].

Suppose that the statue was scanned by two robots from different angles. Let $P = \{p_1, \cdots, p_k\}$ and $Q = \{q_1, \cdots, g_r\}$ denote two sets of 3D points in $\mathbb{R}^3$. Let $L = \{l_{1,} \cdots, l_t\}$ denote a set of 3D points that belong to overlapping areas, where $L \in$ {P,L}. A 3D model of the statue is presented in different colors (Figure 3.11). Each color represents one of two scans, P or Q. The point set L marked with red belongs to both scans, P and Q. The presented model is a ground truth necessary for the analysis of the accuracy of the neural network. Information about the model is presented in Table 3.1.

Table 3.1 The dimensions of two different sets of points, P and Q, of the same model, N_k and N_r, which is visible from different angles and set of poit L, which is belongs to both parts P and Q.

Parts of 3D scan	P Set of points (blue)	Q Set of points (green)	L Overlapping parts (red)
Number of 3D points	$N_k = 15000$	$N_r = 17862$	$N_t = 5000$

As can be seen from 3D points presented in the table, the overlap of the two three-dimensional point clouds is about 30 percent. A set of points $L = \{l_{1,} \cdots, l_t\}$ is a group of training examples, where $l_n = \{p_i, q_j\}$ and it will be used to update the synaptic weights of the neural network.

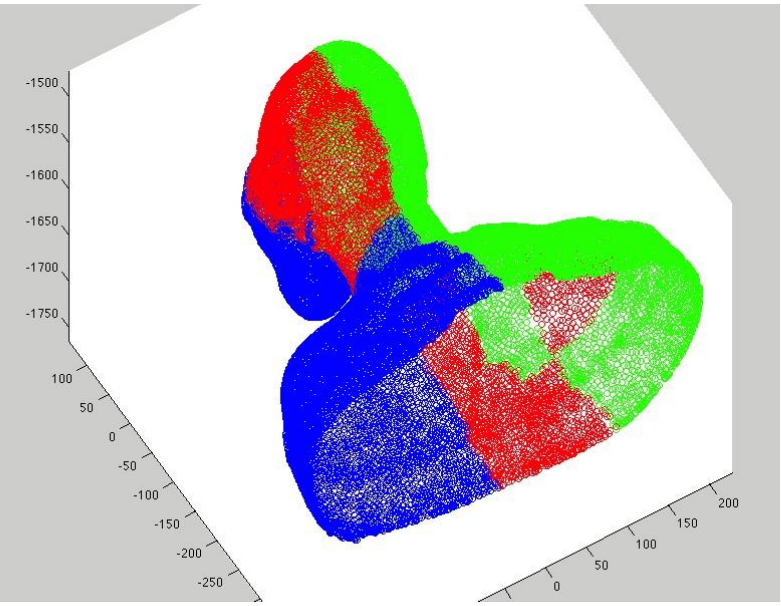


Figure 3.11. 3D mesh of Francesco Laurana female bust. Assumed that the statue was scanned by two robots from different angles. Points visible from both angles are marked with red.

Let $D = \{d_{1,} \cdots, d_r\}$ denote a set of $Q = \{q_1, \cdots, g_r\}$ the transformed point set in $\mathbb{R}^3$, given by the equation

$$d_i = A \cdot q_i + tr, \tag{3.27}$$

where A is a non-singular 3×3 matrix and tr is a transition part of any affine transformation.

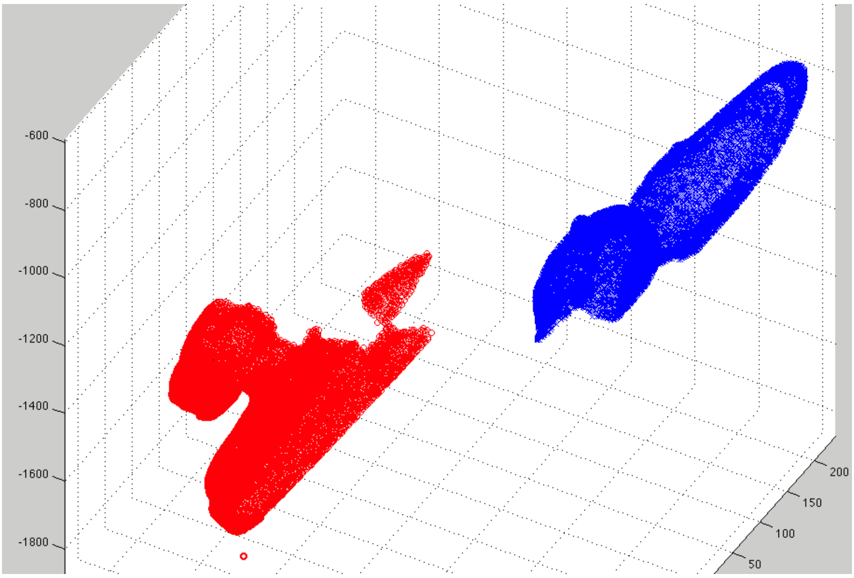


Figure 3.12. Two parts of the same model presented in a common coordinate system after distortion. The new set of 3D points, D is presented in blue.

The affine registration problem (the rigid transformation considered as a special case) can be formulated in terms of affine transformation $A = (A, tr)$ that minimizes the following cost function $E(A, tr)$:

$$dist_i = \sqrt{\sum_{i=1}^{N_r} (q_i - y_i)^2}, \tag{3.28}$$

$$E(A, tr) = \frac{\sum_{i=1}^{N_r} dist_i}{N_r} , \tag{3.29}$$

where y_i – estimated coordinate of the $i\ th$ 3D point, q_i - position of 3D ground truth point, N_r- the number of 3D points.

To solve this problem, the multilayer perceptron was constructed. The network consists of a hidden and an output layer. The hidden layer has 20 neurons. The output layer has three units. The simulation was carried out in Matlab (Figure 3.13).

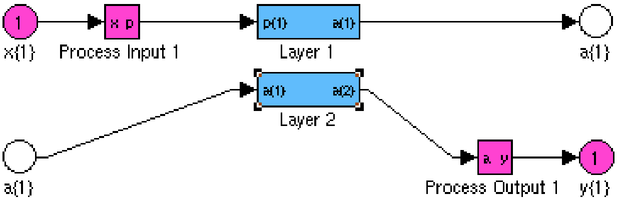


Figure 3.13. Approximation of transformation of an unknown function by a neural network.

In the experiments, the point cloud Q has been transformed several times using different transformation matrices A according to equation (3.27). A neural network is trained on a set of examples from L, where $l_n = \{p_i, d_j\}$, and $d_i = A \cdot q_i + tr$, respectively. A result of the approximation of a transformation function by the neural network is shown in Figure 3.14.

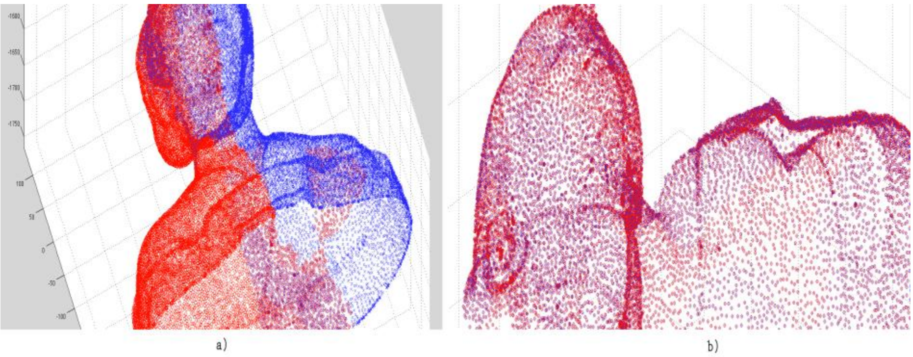


*Figure 3.14. Approximation result of the transformation matrix **A**. The reconstructed model (a), comparison between the reconstructed model and the ground truth (b).*

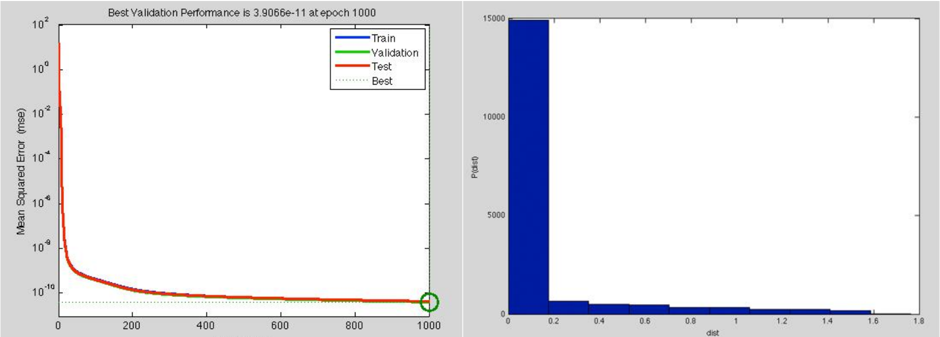
Visual comparison (Figure 3.14 (b)) confirms that by the training process the multilayer perceptron updates synaptic weights for an accurate approximation of A. To compare the accuracy of neural network approximation with that of the ICP method the average value of a transformation error is estimated according to the equation:

$$dist_i = \sqrt{\sum_{i=1}^{N_p} (y_i - gt_i)^2}, \tag{3.30}$$

$$E = \frac{\sum_{i=1}^{N_p} dist_i}{N_p}, \tag{3.31}$$

where E is a mean value of the error function $dist$.

The probability density of the error function $dist$ is presented in Figure 3.15.



a) b)

Figure 3.15. Performance of the neural network (a), the probability density of the error function $\boldsymbol{dist}$ *(b).*

For comparison, the result of the ICP algorithm is presented in Figure 3.16.

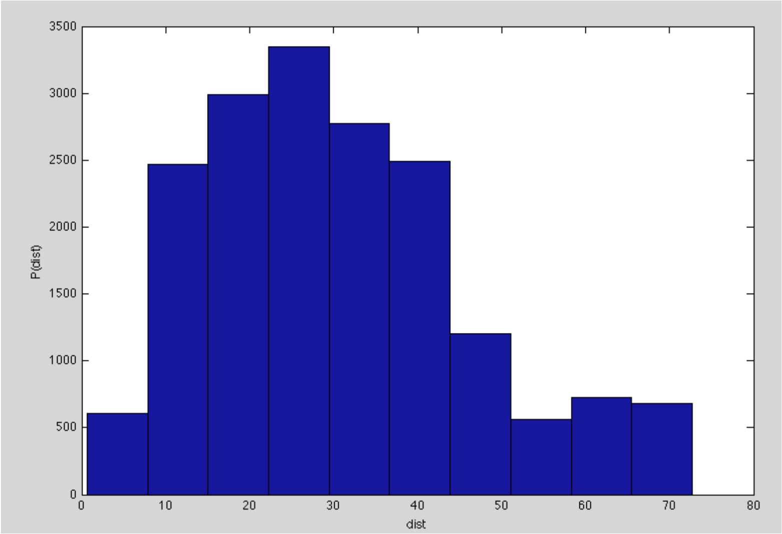


Figure 3.16. The probability density of the error function $\boldsymbol{dist}$ *by using the ICP.*

The mean value of the error function $dist$ could be estimated by equation (3.31) and it equals $E = 30.560$ units for the particular case.

Figure 3.17 summarizes the experimental results. In Table 3.2, the neural network is tested with four different transformation matrixes.

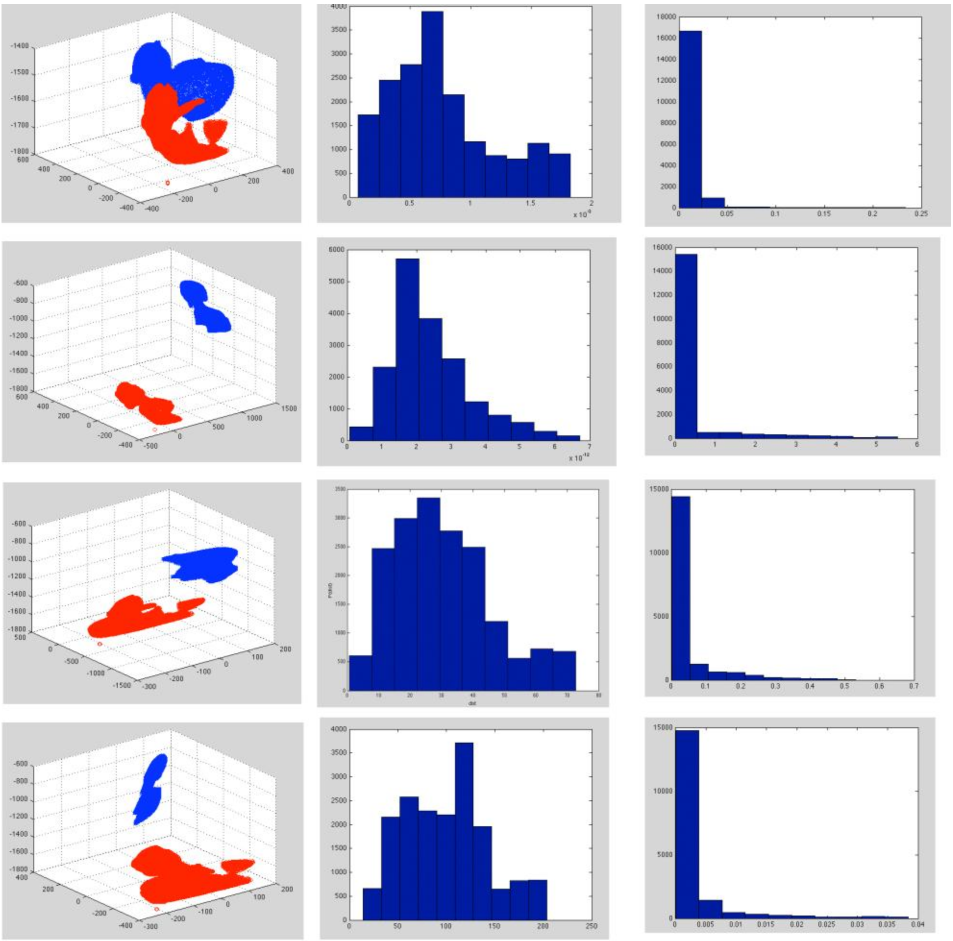


Figure 3.17. Comparison between approximations of several transformation functions. The first column: four different transformations. Second column: probability density of the error function ***dist*** *estimated by the ICP algorithm. Third column: probability density of the error function* ***dist*** *estimated by the neural network.*

Table 3.2. Comparison between the two methods: ICP vs. Neural Network.

Kind of affine transformation $T = \begin{bmatrix} A & tr \\ o^T & 1 \end{bmatrix}$	ICP	Neural Network
	mean value of the error function $dist$ (units)	mean value of the error function $dist$ (units)
Rigid transformation $T_1 = \begin{bmatrix} 0.9997 & -0.0174 & -0.0175 & 100 \\ 0.0171 & 0.9997 & -0.0174 & 300 \\ 0.0178 & 0.0171 & 0.9997 & 70 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$7.67 \cdot 10^{-4}$	0.0059
Rigid transformation $T_2 = \begin{bmatrix} 0.36 & 0.48 & -0.8 & 100 \\ -0.8 & 0.6 & 0 & 300 \\ 0.48 & 0.64 & 0.6 & 70 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$2.42 \cdot 10^{-12}$	0.3381
Rigid transformation $T_3 = \begin{bmatrix} 1 & 0.5 & 0 & 100 \\ 0 & 0.7071 & 0.7071 & 300 \\ 0 & -0.7071 & 0.1072 & 70 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	30.5096	0.0427
Affine transformation $T_4 = \begin{bmatrix} 0.36 & 0.48 & 0 & 100 \\ 0 & .6 & 0 & 300 \\ 0.48 & 0.64 & 0.6 & 70 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	103.54	0.0029

As the table shows, the average value of the error function of the neural network has remained relatively stable for the different types of affine transformations. This confirms the fact that the NN successfully solves the problem of approximating nonlinear functions. In order to evaluate the proposed method, in the next trials, the neural network with different noise settings was tested. The Gaussian noise ($\pm x\%$) was added to each coordinate of every point independently. Results for four levels of noise are shown in Table 3.3. The results are plotted also in Figure 3.18.

Table 3.3 Comparison of different algorithms for varying noise levels

Noise σ	Mean value of the error function $dist$ (units)			
	T_1	T_2	T_3	T_4
0	0.0059	0.3381	0.0427	0.0029
0.5	5.5	2.01	1.42	7.86
1	12.45	13.39	11.65	27.53
2	17.57	27.29	12.38	36.42

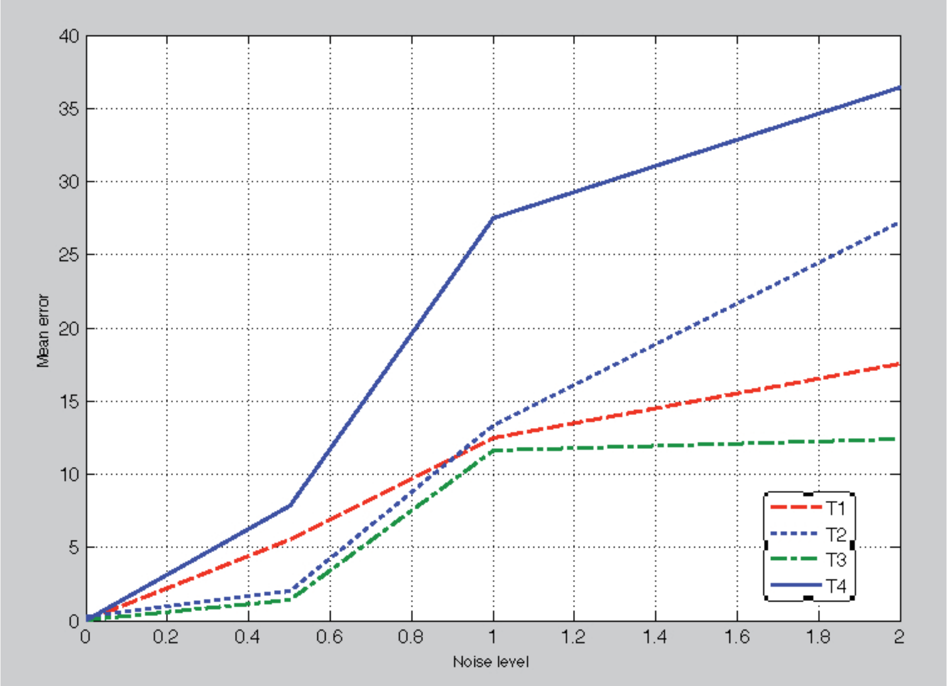


Figure 3.18. Plot of the mean error against the $\boldsymbol{\sigma}$ *of the noise for the transformation matrix from Table 3.1.*

As can be seen, the performance of the neural network degrades as the noise level increases. The properties of the NN for a different number of points in the training sets are the last check point. The result of this trial is presented in Table 3.4 and plotted in Figure 3.19.

Table 3.4 Properties of the NN for different number of points in training set

Number of points	1000	1500	2000	2500	3000	4000	5000
Mean error for T_1	43.54	25.73	5.349	0.57	0.046	0.012	0.0059
Mean error for T_2	22.34	6.038	2.077	0.134	0.024	0.0048	0.0029

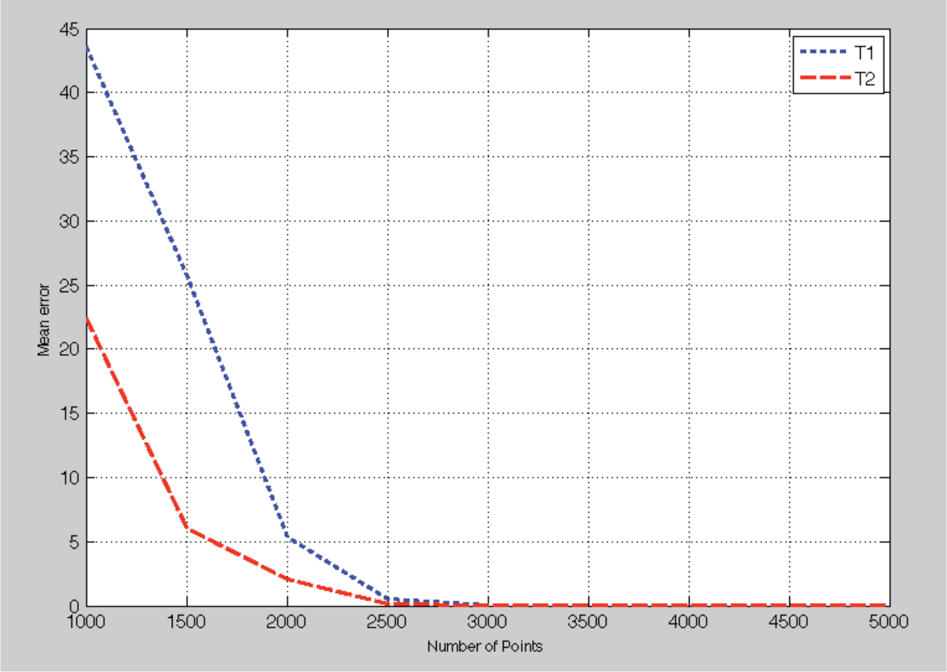


Figure 3.19. Performance of the neural network for different sizes of training sets.

These results explain the importance of proper selection of the data for the training set. If the size of the training set equals 1000 points, and dataset points are selected from a relatively small area, the NN acquires insufficient statistical information for an accurate approximation of the transformation function. The performance of the NN improves linearly with the size of the training data set. To confirm the theory of importance of proper selection of training set points a couple of additional experiments were conducted. The n-number of points were randomly selected from the overlapping area in compliance with one major condition, from different parts of the overlapping area. The property of the NN is presented in Table 3.5.

Table 3.5 The accuracy of the neural network for different sizes of training data sets

Number of points	100	200	400	600	800	1000
Mean error for T_1	5.3872	2.3841	1.4850	1.5750	0.1482	0.1171
Mean error for T_2	0.7314	0.6664	0.5265	0.0637	0.0656	0.0672

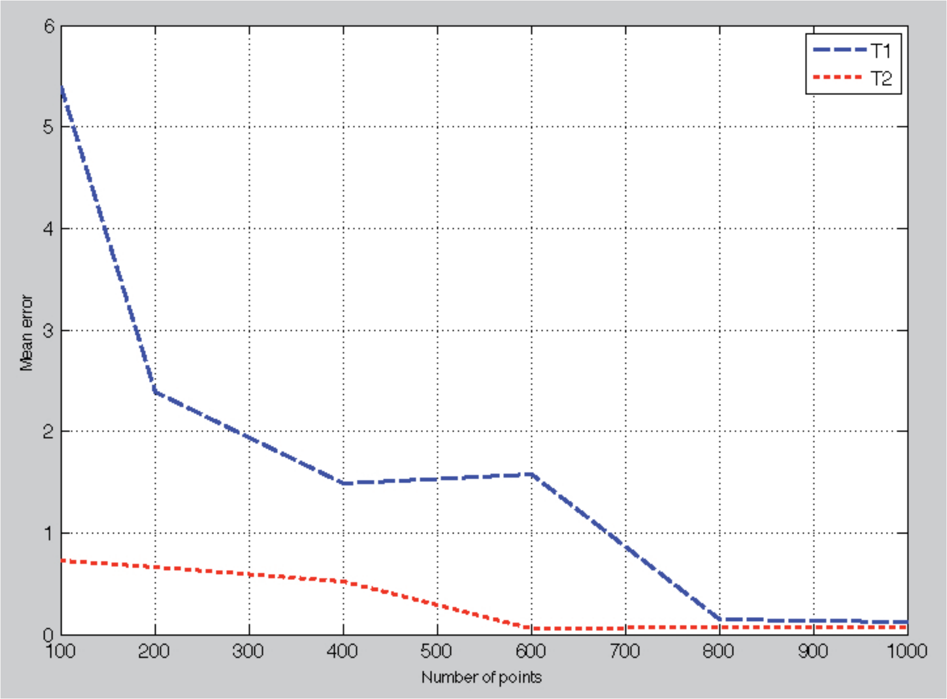


Figure 3.20. The accuracy of the neural network for different examples in the training data set.

Finally, two criteria for stopping of the learning process were defined in the experiment. The learning process was stopped when the error signal $e_k(n)$ (3.9) achieved the desired value. In the experiment the desired value of the error signal was 10^{-12}. The learning process stopped also after 1000 training iterations or epochs. The following option for an automatic completion of the learning process was present; if after 6 epochs the error signals are not reduced any more or stay at the same level, the minimum value of the error signal is achieved.

The performance of the training process for the last experiment is presented in Figure 3.21.

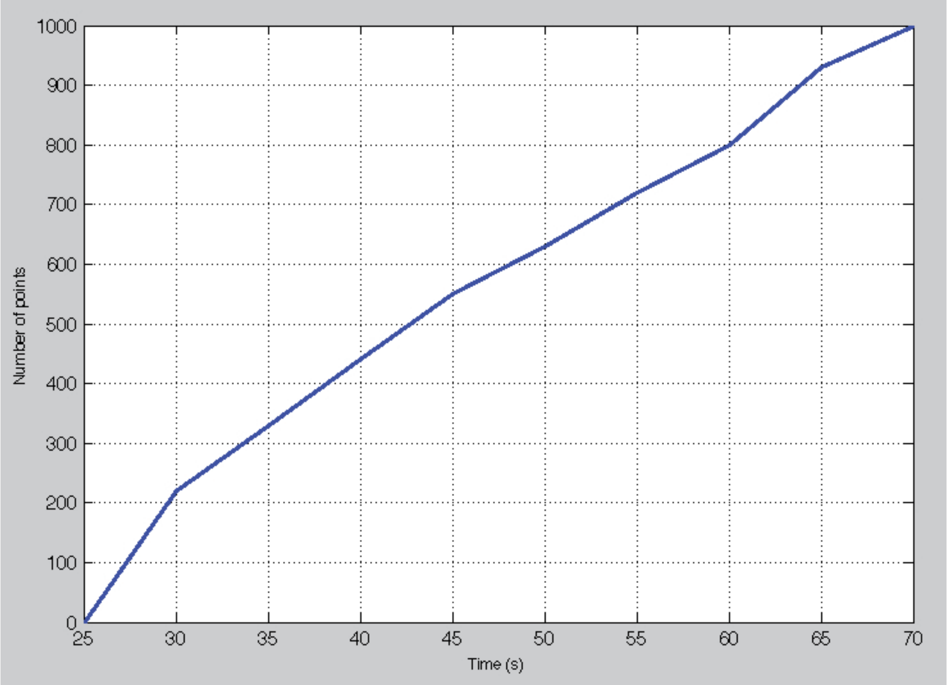


Figure 3.21. Performance of the training process for a multilayer neural network.

The main conclusion from here is that the accuracy of the neural network that approximates the unknown function is highly dependent on the size of the training set and also on the quality of the presented data. It is very important to select points for the training set from different parts of the overlapping area and thereby supplies the NN with maximum amounts of helpful statistical data about an unknown transformation. Figure 3.20 shows that the same mean value of the cost function could be achieved with a smaller size of the training dataset if we follow the proposed strategy by its formation.

3.6 Conclusion

This chapter has described the map-merging problem. If we have two 3D local maps, which have an overlap, then the aim is to estimate an unknown transformation between the maps and create a complete model of 3D space in the common coordinate system. This problem is very similar to the registration problem studied in recent decades. Affine registration in $R^2$and R^3has been extensively investigated. Different kinds of existing methods were studied and their advantages and disadvantages were analyzed.

A new method for solving the registration problem has been proposed here. This method is based on the properties of the neural network to approximate the nonlinear function. A neural network has several advantages over traditional computation methods. The main properties of the neural network were highlighted in this chapter.

To solve the registration problem the multilayer network was chosen. The ability of the neural network and its properties were examined in several experiments. To obtain more statistical information, all experiments were repeated for 100 times. Results of trials are presented in tabular form and plotted in the figures. Tables contain the average values.

Based on the results of the experiment, one main conclusion is obvious. The multilayer neural network is capable of solving the map-merging problem. The approximation of a nonlinear transformation function is robust and could be extended to a n-dimensional case. However, the results are not optimal. The accuracy of the approximation depends on various parameters, such as the structure of the network, the number of neurons in layers, the number of layers, different learning parameters, the number of examples in the training set. Their impact needs further in-depth analysis.

Conclusion

Scientific Results

The current thesis proposes an efficient, rough new method for aligning and merging of 3D local maps into a global map for use in various time and resource critical applications. One of these appilcations serves onboard robot navigation and control tasks. This problem often arises in multi-robot applications. The proposed method involves two steps: detection of overlapping areas between two different maps, alignment of two local maps in a common coordinate system. The new 3D map merging method proposed here will support development of more effective path planning and control approaches for mobile and industrial robots in indoor environments. The proposed solution is to use visual appearance of the images. An image of an observed scene is more informative as compared to any kind of distance sensor. In addition, video sensors are much cheaper. If the overlaps between two maps are established, the affine transformation can be estimated. The geometry relationship between matched images allowed new method to estimate point-to-point correspondences in two image planes. Back transition from 2D to 3D world coordinates gives the point–to-point correspondence between two 3D maps. For the estimation of existing transformation parameters between 3D point clouds, a multilayer neural network is proposed to use in the method.

The alignment of local 3D maps is a complex and challenging problem. The existing methods are briefly covered in Chapter 1. The benefits of landmark-based algorithms are discussed in detail. These algorithms enable problem solution in two steps:

- Detection of existing overlaps
- Estimation of optimal transformation parameters in order to present two local maps in a common coordinate system

However, these methods have a major disadvantage. In order to detect the existing overlaps, local image features are used to compare several images against each other. In general, the process of features processing is very expensive. Due to the high computational complexity, the method works only with short image sequences. In order to improve it, a global descriptor, in this thesis it is a GIST descriptor, is proposed. The descriptor considered as a global feature describes the general properties of a scene.

In Chapter 2 the properties and discriminative power of the GIST descriptor are examined. The global descriptor was included to the image retrieval system in order to solve one of the major problems of autonomous behavior of a mobile robot, the loop-closing problem that the local features are commonly used for. In the proposed method we used the multilayer filtering strategy. The task of each layer is to reject from the image database as many images as possible. The GIST descriptor allowed rejecting 99 percent of the images. Because the discriminative power of the GIST descriptor is low, local features are used in the last filtering step in order to compare the query image with the rest. As result of the introducing new method the required computing time diminished in many times. The experiment showed it is sufficient to check only 0.11 percent of the database images in order to get the winner image. It took about 4 seconds to construct a list of 61 likely candidates in compare to the time performance (14.7189 seconds) for constructing one column of the similarity matrix [20], which is more than 3 time faster. The time performance of the algorithm was tested in Matlab.

Chapter 3 discusses the map-merging problem. Based on the ability of the neural network to approximate a nonlinear function, we examined a very simple multilayer feedforward network with the simplified version of 3D scanned mesh. It is experimentally proved that the neural network is able to successfully solve the map-merging problem and new approach is introduced.

This thesis proposes a new computationally feasible method for online maps alignment and merging. The next section summarizes the work and plans for further investigations.

The Main Scientific Contributions

- Developed a new and effective method for 3D map merging targeted for robot indoor navigation tasks by using visual appearance
- Proposed a new intelligent approach for alignment of local 3D maps created by several robots, in a common coordinate system.
- In order to detect overlaps between several maps (Section 2.5), the image retrieval system based on the multilayer filtering method is presented. The global image representation reduces the computational cost of the method and multilayer-filtering strategy reduces the image retrieval time in about 3 times.
- A novel algorithm for the alignment of two 3D point clouds is presented (Chapter 3). The method proposes to use the neural network for approximation of the affine transformation function.

Future work

During the research, several ideas and problems emerged that require further investigation.

- The image retrieval system discussed in Chapter 2 used a restricted amount of clusters for creation of a database. The next question for future tests is - how many clusters are needed and how their number depends on the number of robots. These questions related to the image retrieval system are still open and should be investigated.
- The experiments carried out in Chapter 3 confirmed the possibility of using artificial neural networks for the alignment of several local maps in a common coordinate system. The influence of the addition of hidden layers on the learning process needs future investigation.
- The accuracy with which the neural network approximates the unknown

transformation function depends on the configuration of points selected for the training set. How to improve the accuracy of mapping is a question to be answered in the future work.
- The research needs to be extended to different environments, especially various outdoor conditions to adjust the method proposed.

Obviously, this is the main vector for further research in this area.

References

1. **Andersen M.R., Jensen T., Lisouski P., Mortensen A.K., Hansen M.K., Gregersen T. and Ahrendt P.** (2012). Kinect Depth Sensor Evaluation for Computer Vision Applications. *Technical report ECE-TR-6.* ISSN: 2245-2087 [Online] http://eng.au.dk/fileadmin/DJF/ENG/PDF-filer/Tekniske_rapporter/Technical_Report_ECE-TR-6-samlet.pdf
2. **Besl P.J., McKay N.D.** (1992). A method for registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256.
3. **Borenstein G.** (2012). Making Things See: 3D vision with Kinect, Processing, Arduino, and MakerBot. ISBN:1449307078
4. **Castellanos J.A., Neira J. and Tardos J.D.** (2001). Multisensor fusion for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 908 – 914.
5. **Chen S., Li Y. F., Zhang J. and Wang W.** (2008). Active Sensor Planning for Multiview Vision Tasks (1st ed.). *Springer Publishing Company, Incorporated*, ISBN:3540770712 9783540770718
6. **Civera J.** (2009). Real-Time EKF-based Structure From Motion. *Universidad de Zaragoza*.
7. **Crowley J.L. and Crowley J.L.** (1989). World modeling and position estimation for a mobile robot using ultrasonic ranging. *IEEE Conference on Robotics and Automation , 3*, pp. 1574–1579.
8. **Davison A.J. Reid I.D, Molton N.D. and Stasse O.** (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 29, Issue 6, pp. 1052-1067.
9. **Dedeoglu G. and Sukhatme G.S.** (2000). Landmark-based matching algorithm for cooperative mapping by autonomous robots. *Distributed Autonomous Robotic Systems* , pp. 251–260.
10. **Donoser M. and Bischof H.** (2006). Efficient maximally stable extreme region (MSER) tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 1, pp. 553-560.
11. **Douze M., Jégou H., Sandhawalia H., Amsaleg L. and Schmid C.** (2009). Evaluation of GIST descriptors for web-scale image search. *Proceedings of the ACM International Conference on Image and Video Retrieval*, Article 19, 8 pages.
12. **Fenwick J.W. and Newman P.M. and Leonard J.J.** (2002). Cooperative concurrent mapping and localization. *IEEE International Conference on Robotics and* Automation, pp. 1810–1817.

13. **Fox D., Burgard W., Kruppa H. and Thrun S.** (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, vol. 8, no. 3, pp. 325-344.
14. **Gelfand N., Mitra N.J., Guibas L.J. and Pottmann H.** (2005). Robust global registration. In *Proceedings of the third Eurographics symposium on Geometry processing*. Article 197.
15. **Gil A., Reinoso O.** (2006) Improving data association in vision-based {SLAM}. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*
16. **Guest G., MacQueen K.M.** (2008). Handbook for Team-Based Qualitative Research ISBN: 978-0-7591-0910-0.
17. **Guizzo E.** (15 May 2013). Kuka Robot Competition Offers 20,000-Euro Award, [Online] http://spectrum.ieee.org/automaton/robotics/industrial-robots/kuka-innovation-mobile-manipulation-award.
18. **Hartley R.I. and Zisserman A.** (2004). Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518.
19. **Haykin S.** (1998). Neural Networks: A Comprehensive Foundation (2nd Edition). *Prentice Hall PTR*, ISBN:0132733501.
20. **Ho K.L., Newman P.** (2005). Multiple Map Intersection Detection using Visual Appearance. *3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems*.
21. **Ho J., Yang M-H., Rangarajan A. and Vemuri B**. (2007). A new affine registration algorithm for matching 2D point sets. *IEEE Workshop on Applications of Computer Vision*, p. 25.
22. **Howard A.** (2006). Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research.* vol. 25, no. 12, pp. 1243-1256.
23. **Inaba M., Katoh N., Imai H.** (1994). Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering: (extended abstract). *Proceedings of the tenth annual symposium on Computational geometry*, pp. 332-339.
24. **Jegou H., Douze M, Schmid C**. (2008). Embedding and weak geometric consistency for large scale image search. *Proceedings of the 10th European Conference on Computer Vision Part I*, pp. 304-317
25. **Jia Z.** (2011), Computer vision with KINECT, [Online] http://www.cs.cornell.edu/courses/CS7670/2011fa/lectures/zhaoyin_kinect.pdf.
26. **Julier S.J. and Jeffrey and Uhlmann K.** (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, pp. 401–422.
27. **Kanungo T., Netanyahu N.S.** (2002). An Efficient k-Means clustering algorithm: analysis and implementation, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7.

28. **Kin Leong Ho, Newman P.** (2006). Loop closure detection in SLAM by combining visual and spatial appearance, *Robotics and Autonomous Systems*, vol. 54, pp. 740 -749.
29. **Klein G. and Murray** D. (2007). Parallel tracking and mapping for small AR workspaces. *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 1-10.
30. **Klein G., Murray D.** (2007). Parallel Tracking and Mapping for Small {AR} Workspaces. *International Symposium on Mixed and Augmented Reality.*
31. **Konolige K. and Fox D., Limketkai B., Ko J., Stewart B.** (2003). Map merging for distributed robot navigation. *Intl. Conf. on Intelligent Robots and Systems* , pp. 212– 217.
32. **Kornienko S.V., Kornienko O.A.** (2006). Artificial self-organization and collective artificial intelligence: on the way from of the individual to society. *From model of behavior to artificial intelligence* p.287 - 343.
33. **LeCun Y.** (1993). Efficient learning and second-order methods. *Neural Information Processing Systems*,[Online] http://www.iro.umontreal.ca/~pift6266/A06/refs/YannNipsTutorial.pdf .
34. **Letzing J.** (20 March 2012). Amazon Adds That Robotic Touch, [Online] http://online.wsj.com/article/SB10001424052702304724404577291903244796214.html.
35. **Liu Yang and Zhang Hong** (2012). Visual loop closure detection with a compact image descriptor. *International Conference on Intelligent Robots and Systems (IROS).* ISBN 978-1-4673-1737-5, pp.1051-1056.
36. **Lowe D.G.** (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, vol. 60, Issue 2, pp. 91-110.
37. **Maji S.** (2010). A Comparison of feature descriptors, [Online] http://ttic.uchicago.edu/~smaji/reports/cs294-6-report.pdf.
38. **Matas J. and Chum O. and Urban M. and Pajdla T.** (2002). Robust wide baseline stereo from maximally stable extremal regions. *British Machine Vision Conference* , pp. 384-393.
39. **Matthies L. and Shafer S.** (1987). Error Modeling in Stereo Navigation, *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 3, pp. 239 - 250.
40. **Matthies L., Shafer S.A.** (1987), Error modeling in stereo navigation, *IEEE Journal of Robotics and Automation.*
41. **MeshLab. (2013).** [Online] http://meshlab.sourceforge.net/.

42. **McCulloch W.S. and Pitts W.** (1988). A logical calculus of the ideas immanent in nervous activity. *Neurocomputing: foundations of research*, James A. Anderson and Edward Rosenfeld (Eds.). MIT Press, Cambridge, MA, USA, pp. 15-27.
43. **Murillo A.C. and Singh G.** (2013). Localization in urban environments using a panoramic gist descriptor. *IEEE Transactions on Robotics,* vol. 29, no. 1, pp. 146-160.
44. **Murphy K.P., Torralba A., Eaton D. and Freeman W.T.** (*2006*), *Toward Category-Level Object Recognition,* vol. 4170 of Lecture Notes in Computer Science, pp. 382-400.
45. **Oliva A. and Torralba A.** (2001). Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Int. J. Comput. Vision,* vol. 42, number 3, pp.145-175.
46. **Paltsev A.S.** (2006). Application of multi-agent technologies to hierarchical EPS state estimation. Proc. of the International Workshop "Electricity Liberalization and Modernization of Power Systems: Risk Assessment and Optimization for Asset Management", Irkutsk, pp.187-192.
47. **Panait L., Luke** S. (2005.) Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, vol. 11, issue 3, pp. 387-434.
48. **Puzicha J., Buhmann J.M., Rubner Y. and Tomasi C.** (1999). Empirical Evaluation of Dissimilarity Measures for Color and Texture. *Proceedings of the International Conference on Computer Vision,* vol. 2. IEEE Computer Society, Washington, DC, USA, 1165-1172.
49. **Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., Moore R., Kipman A. and Blake A.** (2011). Real-time human pose recognition in parts from single depth images. *IEEE Conference on Computer Vision and Pattern Recognition,* pp 1297-1304.
50. **Sivic J. and Zisserman A.** (2003). Video Google: A Text Retrieval Approach to Object Matching in Videos. *IEEE International Conference on Computer Vision*, vol. 2 , pp. 1470-1477.
51. **Smith R.C. and Cheeseman P.** (1986). On the representation and estimation of spatial uncertainly. *International Journal of Robotics Research*, pp. 56–68.
52. **Spletzer J., Das A.K., Fierro R., Taylor C.J., Kumar V., Ostrowski J.P.** (2001). Cooperative localization and control for multi-robot manipulation. *Int. Conf. on Intelligent Robots and Systems*, pp. 631-636 .
53. **Squire David McG., Müller W., Müller H., Raki J.** (1999). Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. *Pattern recognition letters,* pp. 143-149.

54. **Strasdat H., Montiel J.M.M. and Davison A.J. (**2012). Editors choice article: visual SLAM: why filter?, *Image and Vision Computing*, vol. 30, issue 2, pp. 65-77.

55. **Saenko A., Polte G., Musalimov V**.(2012) Potential Application of Fuzzy Logic in Image Processing and Quality Analysis with Optical Measurement Systems. *MTA Review*, vol. 22, no.4, pp.187-196.

56. **Tallinn University of Technology**, **Dept. of Mechatronics**, (2005-2009) "Universal Ground Vehicle", Research project L523 Report.

57. **The Rawseeds prodject** (2013). [Online] http://www.rawseeds.org.

58. **Torralba A., Murphy K.P., Freeman W.T. and Rubin M.A.** (2003). Context-based vision system for place and object recognition. *IEEE International Conference on Computer Vision*, vol. 2, p. 273

59. **Triggs B., McLauchlan P.F., Hartley R.I., and Fitzgibbon A.W.** (1999). Bundle adjustment - a modern synthesis. *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, Bill Triggs, Andrew Zisserman, and Richard Szeliski (Eds.). Springer-Verlag, London, UK, UK, 298-372.

60. **Vailaya Aditya and Jain Anil and Zhang Hong Jiang** (1998). On image classification: city images vs. landscapes. *Pattern Recognition.* vol. 31, pp. 1921-1935.

61. **Wang T., Basu A**. (2006). Automatic estimation of 3D transformations using skeletons for object alignment. In *Proceedings of the 18th International Conference on Pattern*, vol. 1, *IEEE Computer Society*, pp. 51-54.

62. **Widrow B. and Hoff M.E.** (1988). Adaptive switching circuits. *Neurocomputing: foundations of research*, James A. Anderson and Edward Rosenfeld (Eds.), pp. 123-134.

63. **Wolfson H.J. and Rigoutsos I.** (1997). Geometric hashing: an overview. *IEEE Computational Science & Engineering*, vol. 4, issue 4, pp. 10-21.

64. **Zisserman A.** (2013) . MATLAB Functions for Multiple View Geometry. [Online] http://www.robots.ox.ac.uk/~vgg/hzbook/code/.

65. **3D-CoForm Project (2013).** [Online] http://www.3d-coform.eu/

List of Publications

1. **Shvarts, D.; Tamre, M.** (2012). Local and Global Descriptors for Place Recognition in Robotics. *8th International Conf. of DAAAM Baltic, Industrial Engineering,* Tallinn, Estonia, (Eds.) T.Otto. Tallinn, Estonia: Tallinn University of Technology Press, pp. 351 - 356.

2. **Shvarts, D.; Tamre, M**. (2011) The map merging approach for multi-robot monocular SLAMS. *"Recent advanced in mechatronics" by SPRINGER,* pp. 21 - 32.

3. **Shvarts, D.; Tamre, M**. (2009). Методы улучшения изображения для систем технического зрения мобильного робота. *Journal of international Scientific Publications: Materials, Methods & Technologies*, vol. 3, pp. 49-58.

Other publications:

4. **Shvarts, D.; Tamre, M.** (2010). Review of the methods for estimation of 2D homography. *9th International Symposium "Topical Problems In The Field Of Electrical And Power Engineering" and "Doctoral School of Energy and Geotechnology II"*, Pärnu, Estonia, June 14 - 19, pp. 204 - 208. Tallinn University of Technology

5. **Shvarts, D.; Tamre, M.** (2010). Computer vision in applications of adaptive management. *9th International Symposium "Topical Problems In The Field Of Electrical And Power Engineering" and "Doctoral School of Energy and Geotechnology II"***,** Pärnu, Estonia, January 11 - 16, pp. 204 - 208. Tallinn University of Technology

6. **Shvarts, D.; Tamre, M**. (2009). Intelligent Navigation Control of Mobile Robots in Unknown Environment. *7th International Symposium "Topical Problems in the Field of Electrical and Power Engineering",* Doctoral School of Energy and Geotechnology. Narva-Jõesuu, Estonia, 16.06.-19.06, pp. 39 - 42. Estonian Society of Moritz Hermann Jacobi

Abstract

The term navigation is commonly understood as a technology of computing an optimal route. Navigation capabilities of the existing autonomous vehicle are limited by the boundary of the constructed local map. However, in many practical applications, the robot has to operate in different places, often without any prior knowledge about the environment at all. In such situations the robot is unable to navigate effectively. The objective of this thesis is developing a new and effective method for creating a global map from several local maps created by collaborating single robots.

The thesis proposes an intelligent method that allows merging several local maps in a global map. The method uses visual appearance for detection of existed maps' overlaps, which significantly reduces the cost of the system. The artificial neural network is used to align two different local maps in a common coordinate system.

The thesis consists of four parts: review of literature, definition of two subproblems, subproblems detailed analysis, theoretical foundations, practical experiments and conclusions. The introduction presents the objectives of the thesis and describes its structure.

Chapter ”Review of existing methods” describes the characteristics of existing methods for alignment of local maps in a global map. The analysis of the proposed solutions forms the key direction of the research.

The map-merging problem could be split into two subproblems:

- detection of existing maps' overlaps
- alignment the two 3D maps in a common coordinate system

The first subproblem is very similar to the loop-closing problem, which is known from the individual SLAM applications. In SLAM the local properties of the scene play the major role for detection of revisited places. The main scientific novelty of proposed method is using of global descriptor for overlaps detection. The theoretical section of chapter ”Use of global descriptors as an alternative to local descriptors” focuses on analysis of global descriptor properties and brings forth a possible usage scenario. The global descriptor is included to the image retrieval system where local features are traditionally used. The proposed multi-stage filtration procedure based on the comparison of global properties of two different scenes is one of key issues for the method effectiveness. This strategy involves progressive reduction, from layer to layer, of a number of relevant images from the image database. Due to the fact that the use of global descriptor is very fast and compact, the proposed strategy significantly reduces the image retrieval time and increases the number of processed images . The local points matching procedure and the epipolare geometry constraints' verification is incorporated at the final phase of the method in order to determine the winner image from the set of best candidates. After a series of successful experiments with the data obtained from one robot,

it becomes possible to extend this image finding strategy to a multi-robot case, where image database will include a huge number of images.

If the image retrieval system has detected the winner image in the image database it means that between two local maps exist an overlap. The projective geometry technique allows estimating the point-to-point correspondence between image planes of query and winner images and also between points in 2D image plane and correspondent 3D points in world coordinate system. Due to the intensive use of high-accuracy measurement tools for sensing a 3D space in robotics applications, the task of computing correspondence between points in image plane and real 3D coordinates is trivial.

The second subproblem of the map-merging problem considered as the problem of alignment of 3D point clouds in a common coordinate system is a subject of the last chapter “Map-merging problem in multi-robot applications”. The analyses of the existing methods have shown their shortcomings.

The proposed method uses artificial neural network for approximation of the affine transformation function. The properties of the NN have been examined in a number of experiments and compared with the ICP algorithm. A number of trials with the simplified version of 3D scanned mesh of Ippolita Sforza’s statue have shown that the NN is able to successfully solve the map-merging problem. The properties of the NN were tested for different initial conditions, several noise levels and several affine transformations. Despite the fact that the experiments were done using a very simple NN, with one hidden layer, the accuracy of approximation of a nonlinear transformation function is robust and much higher for NN in comparison with ICP algorithm.

The content of this doctoral thesis is summarized in conclusion, which outlines the main achievements and future-oriented ideas.

Kokkuvõte

Termini navigeerimine all mõistetakse tavapäraselt optimaalse teekonna leidmise protseduuri. Autonoomse sõiduki e. roboti navigeerimisvõimekus on piiratud roboti poolt või väliselt loodud lokaalse kaardi piiridega. Praktikas peab robot paljudel juhtudel tegutsema erinevates asukohtades ilma mingisuguse eelinformatsioonita selle asukoha osas. Sellises situatsioonis on roboti tegutsemise efektiivsus oluliselt piiratud. Käesoleva dissertatsiooni eesmärk on metoodika väljatöötamine mitmetest üksikkaartidest 3D globaalse kaardi kokkupanemiseks, mis oleks kasutatav näiteks mitmesuguste robootika rakendustes.

Dissertatsioonis on pakutud välja mitme lokaalkaardi üheks globaalkaardiks ühendamise uus metoodika. Väljatöötatud metoodika kasutab visuaalse sarnasuse kriteeriume olemasolevate kaartide ülekatte määramiseks, mis kiirendab ja lihtsustab oluliselt selliste süsteemide tööd ning võimaldab uut lahendust kasutada otse roboti peal. Lokaalsete kaartide ühitamiseks ühtsesse koordinaatsüsteemi kasutab uus metoodika tehisnärvivõrku.

Dissertatsioon koosneb üldiselt neljast osast: kirjanduse ülevaade; alamprobleemide defineerimine ja nende detailne analüüs koos lahenduste väljapakkumisega; praktilised eksperimendid ja järeldused. Sissejuhatuses on esitatud dissertatsiooni eesmärgid ja kirjeldatud töö struktuuri.

Töö esimeses osas on kirjeldatud seni kasutatavate lokaalsete kaartide üheks globaalkaardiks ühendamise metoodikaid. Kasutatavate metoodikate analüüsi alusel on selles dissertatsioonis formuleeritud põhilised suunad edasiseks uurimistööks.

Kaartide ühendamise ülesanne on jagatav kaheks järgnevaks alamülesandeks:

- kaartide ülekatte tuvastamine;
- kahe 3D kaardi joondamine ühises koordinaatsüsteemis.

Esimene alamülesanne on väga sarnane tsükli sulgemisprobleemiga individual-SLAM rakendustes. SLAM puhul mängivad tuttava koha tuvastamisel olulisimat rolli süsteemi lokaalsed omadused. Väljapakutud metoodika olulisim teaduslik uudsus on globaaldeskriptori kasutusele võtmine ülekatete tuvastamisel. Järgnev osa keskendub globaaldeskriptori omaduste analüüsile ja esitab selle võimaliku kasutamisstsenaariumi uues metoodikas. Globaaldeskriptorit kasutatakse kujutise otsingualgoritmis traditsiooniliselt kasutatavate lokaaltunnuste asemel. Mitmeetapilise filtreerimise protseduur, mis baseerub kahe erineva stseeni globaalomaduste võrdlemisel, võimaldas selles töös saavutada uue metoodika oluliselt suurema efektiivsuse. Uue metoodika strateegia sisaldab järk-järgulist, kiht kihilt oluliste andmebaasi kujutiste arvu vähendamist. Kuna globaaldeskriptori kasutamine on väga kiire ja kompaktne protseduur, siis kiirendab väljapakutud lahendus oluliselt kujutise otsimise protseduuri ja aitab suurendada analüüsitavate kujutiste arvu. Lokaalpunktide võrdlemise protseduur ja epipolaargeomeetria piirangute verifitseerimine on ühendatud uue meetodi lõppfaasis, et määrata parim kujutis kandidaatkujutiste kogumist. Pärast ühelt robotilt saadud andmete töötlemist on võimalik laiendada seda kujutise strateegiat robotperele, kus kujutiste andmebaas sisaldab juba väga suurt hulka kujutisi.

Kui kujutise leidmise algoritm leiab parima kujutise kujutiste andmebaasist, siis tähendab see, et kahe lokaalse kaardi vahel eksisteerib ülekate. Projektiivse geomeetria abil on võimalik hinnata punkt punktilt päringu kujutise ja parima vastava kujutise vastavust ja samuti vastavust punktide vahel 2D kujutise tasandil ning vastavate 3D punktide kokkulangevust üldises koordinaatsüsteemis. Kuna robootikas kasutatakse laialt suhteliselt täpseid seadmeid robotit ümbritseva 3D ruumi tunnetamiseks, siis on ülesanne leida punktide vastavus kujutise tasandil ja tegelikes 3D koordinaatides triviaalne.

Teine alamprobleem kaartide ühendamisel on 3D punktiparve orienteerimine e. joondamine üldises koordinaatsüsteemis. Seni olemasolevate meetodite analüüs osutab nende olemasolevate meetodite mitmetele puudustele.

Uus, selles töös väljapakutud meetod kasutab tehisnärvivõrku, et aproksimeerida afiinset transformatsiooni funktsiooni. Närvivõrgu omadusi on testitud paljudes katsetes ja võrreldud ka ICP algoritmiga. Rida katseid Ippolita Sforza kuju lihtsustatud skaneeringu punktiparvega näitavad, et närvivõrk on võimeline edukalt lahendama kaartide ühendamise probleemi. Närvivõrgu omadusi on testitud erinevates lähtetingimustes, erineva müra korral ja erinevate afiinsete transformatsioonide puhul. Hoolimata asjaolust, et katsed on tehtud väga lihtsa närvivõrgu abil, kus on ainult üks varjatud kiht, on mittelineaarse funktsiooni teisenduse lähenduse täpsus väga häirekindel ja see täpsus on oluliselt kõrgem närvivõrgu puhul, võrreldes seni kasutatava ICP algoritmiga.

Dissertatsiooni sisu on kokku võetud viimases osas, kus tuuakse välja põhilised tulemused ja tulevase töö suunad.

Elulookirjeldus

1. Isikuandmed

Ees- ja perekonnanimi	Dmitry Shvarts
Sünniaeg ja -koht	19. nov 1973, Stavropol
Kodakondsus	Vene

2. Kontaktandmed

Aadress	Olevi 31-1, 30325, Kohtla-Järve, Eesti
Telefon	+372 58509613
E-post	shvarts.dmitry@gmail.com

3. Hariduskäik

Õppeasutus (nimetus lõpetamise ajal)	Lõpetamise aeg	Eriala/kraad
Stavropoli Kõrgem Sõjakool	1990–1995	Automaatika, juhtimis- ja sidesüsteemide insener

4. Keelteoskus

Vene keel	kõrgtase
Saksa keel	kesktase
Inglise keel	kesktase

5. Teenistuskäik

Töötamise aeg	**Ülikooli, teadusasutuse või muu organisatsiooni nimetus**	**Ametikoht**
01.09.2009 – …	TTÜ Virumaa Kolledž, ehituse ja mehaanika lektoraat	lektor
01.09.2007 – 31.08.2009	TTÜ Virumaa Kolledž, ehituse ja mehaanika lektoraat	assistent
2005 – 2007	TTÜ Virumaa Kolledž	tehnikaspetsialist

Curriculum Vitae

1. Personal data

Name	Dmitry Shvarts
Date and place of birth	19 Nov 1973, Stavropol

2. Contact information

Address	Olevi 31-1, 30325, Kohtla-Järve, Estonia
Phone	+372 58509613
E-mail	shvarts.dmitry@gmail.com

3. Education

Institution	Graduation Year	Field of study/degree
Stavropol Higher Military School of Communication	1990–1995	Automatic control and communication systems/engineer

4. Language competence/skills

Russian	Fluent
English	Intermediate
German	Intermediate

5. Professional Employments

Working period	Name of University, Research Institution or other organization	Position
01.09.2009 –	TUT Virumaa College, Construction and Mechanics Division	lecturer
2007 – 2009	TUT Virumaa College, Construction and Mechanics Division	assistant
2006 – 2007	TUT Virumaa College	technical specialist

DISSERTATIONS DEFENDED AT TALLINN UNIVERSITY OF TECHNOLOGY ON *MECHANICAL ENGINEERING*

1. **Jakob Kübarsepp**. Steel-Bonded Hardmetals. 1992.
2. **Jakub Kõo**. Determination of Residual Stresses in Coatings & Coated Parts. 1994.
3. **Mart Tamre**. Tribocharacteristics of Journal Bearings Unlocated Axis. 1995.
4. **Paul Kallas**. Abrasive Erosion of Powder Materials. 1996.
5. **Jüri Pirso**. Titanium and Chromium Carbide Based Cermets. 1996.
6. **Heinrich Reshetnyak**. Hard Metals Serviceability in Sheet Metal Forming Operations. 1996.
7. **Arvi Kruusing**. Magnetic Microdevices and Their Fabrication methods. 1997.
8. **Roberto Carmona Davila**. Some Contributions to the Quality Control in Motor Car Industry. 1999.
9. **Harri Annuka**. Characterization and Application of TiC-Based Iron Alloys Bonded Cermets. 1999.
10. **Irina Hussainova**. Investigation of Particle-Wall Collision and Erosion Prediction. 1999.
11. **Edi Kulderknup**. Reliability and Uncertainty of Quality Measurement. 2000.
12. **Vitali Podgurski**. Laser Ablation and Thermal Evaporation of Thin Films and Structures. 2001.
13. **Igor Penkov**. Strength Investigation of Threaded Joints Under Static and Dynamic Loading. 2001.
14. **Martin Eerme**. Structural Modelling of Engineering Products and Realisation of Computer-Based Environment for Product Development. 2001.
15. **Toivo Tähemaa**. Assurance of Synergy and Competitive Dependability at Non-Safety-Critical Mechatronics Systems design. 2002.
16. **Jüri Resev**. Virtual Differential as Torque Distribution Control Unit in Automotive Propulsion Systems. 2002.
17. **Toomas Pihl**. Powder Coatings for Abrasive Wear. 2002.
18. **Sergei Letunovitš**. Tribology of Fine-Grained Cermets. 2003.
19. **Tatyana Karaulova**. Development of the Modelling Tool for the Analysis of the Production Process and its Entities for the SME. 2004.
20. **Grigori Nekrassov**. Development of an Intelligent Integrated Environment for Computer. 2004.
21. **Sergei Zimakov**. Novel Wear Resistant WC-Based Thermal Sprayed Coatings. 2004.

22. **Irina Preis**. Fatigue Performance and Mechanical Reliability of Cemented Carbides. 2004.

23. **Medhat Hussainov**. Effect of Solid Particles on Turbulence of Gas in Two-Phase Flows. 2005.

24. **Frid Kaljas**. Synergy-Based Approach to Design of the Interdisciplinary Systems. 2005.

25. **Dmitri Neshumayev**. Experimental and Numerical Investigation of Combined Heat Transfer Enhancement Technique in Gas-Heated Channels. 2005.

26. **Renno Veinthal**. Characterization and Modelling of Erosion Wear of Powder Composite Materials and Coatings. 2005.

27. **Sergei Tisler**. Deposition of Solid Particles from Aerosol Flow in Laminar Flat-Plate Boundary Layer. 2006.

28. **Tauno Otto**. Models for Monitoring of Technological Processes and Production Systems. 2006.

29. **Maksim Antonov**. Assessment of Cermets Performance in Aggressive Media. 2006.

30. **Tatjana Barashkova**. Research of the Effect of Correlation at the Measurement of Alternating Voltage. 2006.

31. **Jaan Kers**. Recycling of Composite Plastics. 2006.

32. **Raivo Sell**. Model Based Mechatronic Systems Modeling Methodology in Conceptual Design Stage. 2007.

33. **Hans Rämmal**. Experimental Methods for Sound Propagation Studies in Automotive Duct Systems. 2007.

34. **Meelis Pohlak**. Rapid Prototyping of Sheet Metal Components with Incremental Sheet Forming Technology. 2007.

35. **Priidu Peetsalu**. Microstructural Aspects of Thermal Sprayed WC-Co Coatings and Ni-Cr Coated Steels. 2007.

36. **Lauri Kollo**. Sinter/HIP Technology of TiC-Based Cermets. 2007.

37. **Andrei Dedov**. Assessment of Metal Condition and Remaining Life of In-service Power Plant Components Operating at High Temperature. 2007.

38. **Fjodor Sergejev**. Investigation of the Fatigue Mechanics Aspects of PM Hardmetals and Cermets. 2007.

39. **Eduard Ševtšenko**. Intelligent Decision Support System for the Network of Collaborative SME-s. 2007.

40. **Rünno Lumiste**. Networks and Innovation in Machinery and Electronics Industry and Enterprises (Estonian Case Studies). 2008.

41. **Kristo Karjust**. Integrated Product Development and Production Technology of Large Composite Plastic Products. 2008.

42. **Mart Saarna**. Fatigue Characteristics of PM Steels. 2008.

43. **Eduard Kimmari**. Exothermically Synthesized B_4C-Al Composites for Dry Sliding. 2008.

44. **Indrek Abiline**. Calibration Methods of Coating Thickness Gauges. 2008.

45. **Tiit Hindreus**. Synergy-Based Approach to Quality Assurance. 2009.

46. **Karl Raba**. Uncertainty Focused Product Improvement Models. 2009.

47. **Riho Tarbe**. Abrasive Impact Wear: Tester, Wear and Grindability Studies. 2009.

48. **Kristjan Juhani**. Reactive Sintered Chromium and Titanium Carbide-Based Cermets. 2009.

49. **Nadežda Dementjeva**. Energy Planning Model Analysis and Their Adaptability for Estonian Energy Sector. 2009.

50. **Igor Krupenski**. Numerical Simulation of Two-Phase Turbulent Flows in Ash Circulating Fluidized Bed. 2010.

51. **Aleksandr Hlebnikov**. The Analysis of Efficiency and Optimization of District Heating Networks in Estonia. 2010.

52. **Andres Petritšenko**. Vibration of Ladder Frames. 2010.

53. **Renee Joost**. Novel Methods for Hardmetal Production and Recycling. 2010.

54. **Andre Gregor**. Hard PVD Coatings for Tooling. 2010.

55. **Tõnu Roosaar**. Wear Performance of WC- and TiC-Based Ceramic-Metallic Composites. 2010.

56. **Alina Sivitski**. Sliding Wear of PVD Hard Coatings: Fatigue and Measurement Aspects. 2010.

57. **Sergei Kramanenko**. Fractal Approach for Multiple Project Management in Manufacturing Enterprises. 2010.

58. **Eduard Latõsov**. Model for the Analysis of Combined Heat and Power Production. 2011.

59. **Jürgen Riim**. Calibration Methods of Coating Thickness Standards. 2011.

60. **Andrei Surzhenkov**. Duplex Treatment of Steel Surface. 2011.

61. **Steffen Dahms**. Diffusion Welding of Different Materials. 2011.

62. **Birthe Matsi**. Research of Innovation Capasity Monitoring Methodology for Engineering Industry. 2011.

63. **Peeter Ross**. Data Sharing and Shared Workflow in Medical Imaging. 2011.

64. **Siim Link**. Reactivity of Woody and Herbaceous Biomass Chars. 2011.

65. **Kristjan Plamus**. The Impact of Oil Shale Calorific Value on CFB Boiler Thermal Efficiency and Environment. 2012.

66. **Aleksei Tšinjan**. Performance of Tool Materials in Blanking. 2012.

67. **Martinš Sarkans**. Synergy Deployment at Early Evaluation of Modularity of the Multi-Agent Production Systems. 2012.

68. **Sven Seiler**. Laboratory as a Service – A Holistic Framework for Remote and Virtual Labs. 2012.

69. **Tarmo Velsker**. Design Optimization of Steel and Glass Structures. 2012.

70. **Madis Tiik**. Access Rights and Organizational Management in Implementation of Estonian Electronic Health Record System. 2012.

71. **Marina Kostina**. Reliability Management of Manufacturing Processes in Machinery Enterprises. 2012.

72. **Robert Hudjakov**. Long-Range Navigation for Unmanned Off-Road Ground Vehicle. 2012.