

Robust Real-time Landmark Recognition for Humanoid Robot Navigation

Mohammed Elmogy and Jianwei Zhang
TAMS, Department of Informatics
University of Hamburg
Vogt-Kölln-Straße 30, D-22527 Hamburg, Germany
{elmogy, zhang}@informatik.uni-hamburg.de

Abstract - Landmark recognition is identified as one important research area in robot navigation systems. It is a key feature for building robots capable of navigating and performing tasks in human environments. However, current object recognition research largely ignores the problems that the mobile robot context introduces.

We developed a landmark recognition system which is used by a humanoid robot to identify landmarks during its navigation. The humanoid landmark recognition system is based on a two-step classification stage which is robust and invariant towards scaling and translations. Also, it provides a good balance between fast processing time and high detection accuracy. An appearance-based classification method is initially used to provide the rough initial estimate of the landmark. It is followed by a refinement step using a model-based method to estimate an accurate classification of the object. The goal of our work is to develop a rapid, robust object recognition system with a high detection rate that can actually be used by a humanoid robot to recognize landmarks during its navigation.

Index Terms – Landmark recognition, appearance-based recognition, model-based recognition, robot vision.

I. INTRODUCTION

Mobile robots have been widely used in various application areas such as space missions, military operations, personal assistants to humans, cleaning, entertainment, and tour guiding. The great variety of mobile robots' applications forces researchers to create and develop reliable and efficient robotics systems. Many key research problems of mobile robot applications should be considered in the development of robot architecture design and modules such as navigation, localization, vision-based recognition, speech recognition, and dialog processing [1]. On the other hand, some human-robot interaction (HRI) mechanisms and mobile robot navigation techniques should be incorporated in many applications. HRI has been extensively studied by many research groups. It can be categorized as *active* and *passive* HRI [2]. In active HRI, a user actively interacts with a mobile robot via unnatural communication tools such as a joystick or a Graphical User Interface (GUI). In passive interaction, the robot enables the user to use more natural interaction means such as speech or gestures. Therefore, the human user behaves naturally with the mobile robot, as if he were interacting with another person.

Human robot interactions take place in three main modes: manual, autonomous, and semi-autonomous modes [2]. In the manual mode, the user can determine the objects of interest and these objects are recognized automatically by the robot. If the robot fails to recognize the object, then it asks the user for help. The user can interact with the robot vocally via the robot's speech recognition capability or by using any other communication tool. In the autonomous mode, the robot automatically detects the landmarks that have salient features. It then records images of the landmark from different perspectives for object recognition. In the semi-autonomous mode, the robot identifies some potential objects of interest that are distinguishable in the environment and asks if the user is interested in these landmarks or not.

From the perspective of object recognition techniques, robots lack lightweight object perception methods that allow them to interact with their surrounding environment. In order to make robots useful assistants to people in everyday life, the ability to learn and recognize objects is of essential importance. Object recognition in real scenes is one of the challenging problems in computer vision, as it is necessary to deal with difficulties such as viewpoint changes, occlusions, illumination variations, background clutter, or sensor noise [3]. Furthermore, in a mobile robotics scenario a new challenge is added to the list: computational complexity. All these complications make object recognition in real scenes a difficult problem that will demand a significant research effort in the coming years.

Object recognition in robot navigation is used to detect and classify landmarks during navigation. It is also used to localize the current position of the robot with respect to the detected landmarks. This process is called the robot's self localization. It is defined as the process of estimating the initial position of the robot with respect to a global coordinate system or according to recognized landmarks. Landmark-based localization techniques are common in robotics. They can be classified into active and passive landmarks. Active landmarks are captured and transmitted to the robot for sensing and analysis, such as images acquired by the robot's cameras. Passive landmarks are detected by the robot without any transmitted signals, such as the output of a laser sensor [2].

In general, approaches to solving the recognition problem can be classified into two categories: appearance-

based (or so-called global) methods, and model-based (or so-called local) methods. Appearance-based methods are based on the overall visual appearance of the object. They often represent the object with a histogram of certain features extracted during the training process, such as a color histogram which represents the distribution of object colors. Whereas model-based methods rely on specific geometric features of the object such as small texture patches or particular features. For the robot to recognize an object, the object must appear large enough in the camera image. If the object is too small, local features cannot be extracted from it. Global appearance-based methods also fail to recognize the object, since the size of the object is small in relation to the background, which commonly results in a high number of false positives. A more natural approach in terms of computational efficiency is the use of appearance-based methods for providing a rough initial estimate followed by a refinement step using model-based methods, to estimate the full pose of the object [4, 5]. In addition, this proposed method shows a significant robustness with respect to scaling and translations.

The rest of this paper is organized as follows: The next section discusses the current work on robot object recognition. It also presents some current object recognition techniques which are used in mobile robot applications. Section three presents our humanoid landmark recognition system. Its architecture and components are explained in detail. Finally, section four presents the conclusion and feature work.

II. ROBOT OBJECT RECOGNITION METHODS

Object recognition is one of the main research topics in the field of computer vision. However, most methods typically assume that the object is either already segmented from the background or that it occupies a large portion of the image. In robotic applications, there is often a need for a system that can locate objects in the environment. These methods are not valid anymore since the distance to the object and thus its size in the image can vary significantly. Therefore, the robot has to be able to handle and detect objects even when they occupy a very small part of the image. This requires a method that evaluates different parts of the image when searching for an object. This step is called the object detection stage. The object detection method is especially suitable for detecting objects in natural scenes, if it is able to cope with problems such as complex background, varying illumination and object occlusion.

Object recognition algorithms are typically designed to classify objects into one of several predefined classes assuming that the segmentation of the object has already been performed. In general, object detection tasks are much more difficult. Their purpose is to search for a specific object in an image while not knowing beforehand if the object is present in the image or not. Most of the object recognition algorithms may be used for object detection by scanning the image for the object. Regarding the computational complexity, some methods are more suitable

for searching than others. Numerous methods for object recognition have been developed over the last decades, but few of them actually scale to the demands posed by a mobile robotics scenario. Furthermore, most of them concentrate on specific cases. In the remaining part of this section, we will represent some recent object and landmark recognition methods which are used and suitable to mobile robotics applications.

A. Scale Invariant Features Transform

Lowe [3, 6] proposed an object recognition method that uses Scale Invariant Features Transform (SIFT). SIFT is an approach for detecting and extracting local feature descriptors that are reasonably invariant to changes in rotation, scaling, small changes in viewpoint, illumination, and image noise [7]. This object recognition approach is a single-view object detection and recognition system with some interesting characteristics for mobile robots, most significant of which is the ability to detect and recognize several objects at the same time in an un-segmented image. Another interesting feature is the Best-Bin-First algorithm used for approximating fast matching, which reduces the search time by two orders of magnitude. SIFT can be used to detect and classify the objects in real-time. Its average recognition time is approximately one second. On the other hand, this algorithm has two significant drawbacks. First, it performs poorly when recognizing sparsely textured objects [8]. Second, with repetitive textures of the methods based on local features such as SIFT, which can only find the reliable features when the object occupies a significant part of the image. It is very hard to recognize objects that are far away from the camera [9].

B. Bag of Features Approach

On the other hand, Nistér and Stewénius [10] developed the bag of features approach to recognizing segmented objects. This algorithm comes from the text categorization domain, where the occurrence of certain words in documents is recorded and used to train classifiers that can later recognize the subject of new texts. A histogram of descriptor occurrences is built to characterize an image. In order to limit the size of the histogram, a code-book or vocabulary computed by applying a clustering method to the training descriptors is used. A hierarchical vocabulary tree is used, as it allows the coding of a larger number of visual features and simultaneously the reduction of the look-up time in proportion to the number of leaves. The vocabulary tree is built using hierarchical k-means clustering [11], where the parameter k defines the branch factor of the tree instead of the final number of clusters. One of the drawbacks of the bag of features method is that it is designed to work with an accurate segmentation stage prior to classification which can be very time consuming [8]. The second drawback is that if one image contains background with a value greater than a certain threshold, the probability of miss-classification increases. The third drawback is that if

a particular image contains two objects, there is no way to recognize both [3].

C. Color Histograms

In mobile robot research, there are many algorithms which are used to detect and recognize objects by using their color histograms [1, 12, 13]. Ekvall et. al. [1, 4, 5] used a representation called Receptive Field Cooccurrence Histograms (RFCH), where each pixel is represented by a combination of its color and response to different image filters. Thus, the cooccurrence of certain filter responses within a specific radius in the image serves as information basis for building the representation of the object. The specific goal for object detection is an on-line learning scheme that is effective after just one training example, but still has the ability to improve its performance if given time and new examples. In their framework, training images are generated from human demonstrations in two steps. First, the robot captures an image of the scene without the object being present in it. Then the operator places an object in front of the camera and the object is separated using image differentiating. The resulting image, which contains the proposed object, is represented by its color histogram. At the query stage, the objects are identified by matching a color histogram from an image region with a color histogram from a sample of the object using histogram intersection. It has been shown that this method is robust to changes in the orientation, scale, partial occlusion and changes of the viewing position. However, the main drawback of this method is its sensitivity to lighting conditions.

III. HUMANOID LANDMARK RECOGNITION SYSTEM (HLRS)

In this section, we will present our humanoid landmark recognition system (HLRS) which is used to detect and classify different landmarks during the humanoid robot navigation. Our main goal is to implement a robust, accurate, and real-time landmark recognition method. The robot will use a topological map which is generated from the route instructions to decide which landmark will be processed during the navigation [14]. The topological route map is combined with the recognized landmarks to plan the robot's navigational path.

A. Experimental Platform

Our experimental platform is the second generation of Fujitsu's Humanoid for Open Architecture Platform (HOAP-2) [16]. HOAP-2 weighs 7 kg and is 50 cm tall. It is equipped with 25 servo actuators. For feedback purposes, there are four pressure sensors on the bottom of each foot, and there is an accelerometer and gyroscope inside the torso. HOAP-2 is equipped with two 0.25" CMOS unsynchronized cameras [15].

We chose a humanoid robot for our implementation because human interaction with robots is easier if robots have a humanoid shape. The second reason is that humans will more readily accept robots with a humanoid shape. And

finally, the efficiency of teaching and programming a robot is highest with humanoids.

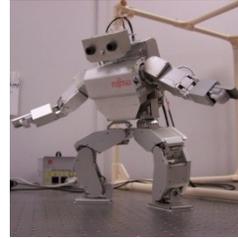


Fig. 1. HOAP-2 Humanoid.

We used the HOAP-2 humanoid robot to detect and recognize landmarks during a navigation task. It is equipped with two Logitech web quick-cameras. These cameras need an initial 10 – 15 seconds to focus and remove the blurring effect, which can cause the robot to miss the objects. The captured images have a resolution of 320 x 240 pixels. The recognition process was carried out on a 1.73 GHz Duo processor laptop with 1 GB RAM.

B. HLRS Architecture

We have developed a landmark recognition system that is effectively based on a two-step classification process. An appearance-based classification method is first used to get a fast and rough estimation of the landmarks. The resulting hypotheses are refined by a model-based classification method to get accurate landmark recognition. The combination between these two methods provides a computational efficiency procedure. The color histogram of the detected landmark provides the rough initial estimate of the landmark. It is followed by a refinement step using SIFT to get an accurate estimation for the landmark. On the other hand, we used the disparity map combined with recognized landmarks to calculate the landmark nearest to the robot. The position of this landmark in the real world is determined by calculating the triangulation. Fig. 2 shows the framework of HLRS. It is divided into three main parts. First, the calibration and triangulation stage that is responsible for determining camera parameters and also the position of the landmark in the real world. Second, the stereo vision stage that is responsible for creating a disparity map between the two cameras and is also used to detect the landmark nearest to the robot. The last stage is the landmark classification stage which is responsible for the detection and recognition of the landmarks from the captured frame. In the following sections, the main building blocks of HLRS will be discussed in detail.

C. Camera Calibration and Triangulation

Camera calibration is the task of relating the ideal pinhole model of the camera to an actual imaging device (internal calibration) and of retrieving the relative position and orientation of the cameras (external calibration) [16]. We used camera calibration to determine the geometry of the stereo setting which is needed for triangulation and also for removing radial distortions provided by camera lenses.

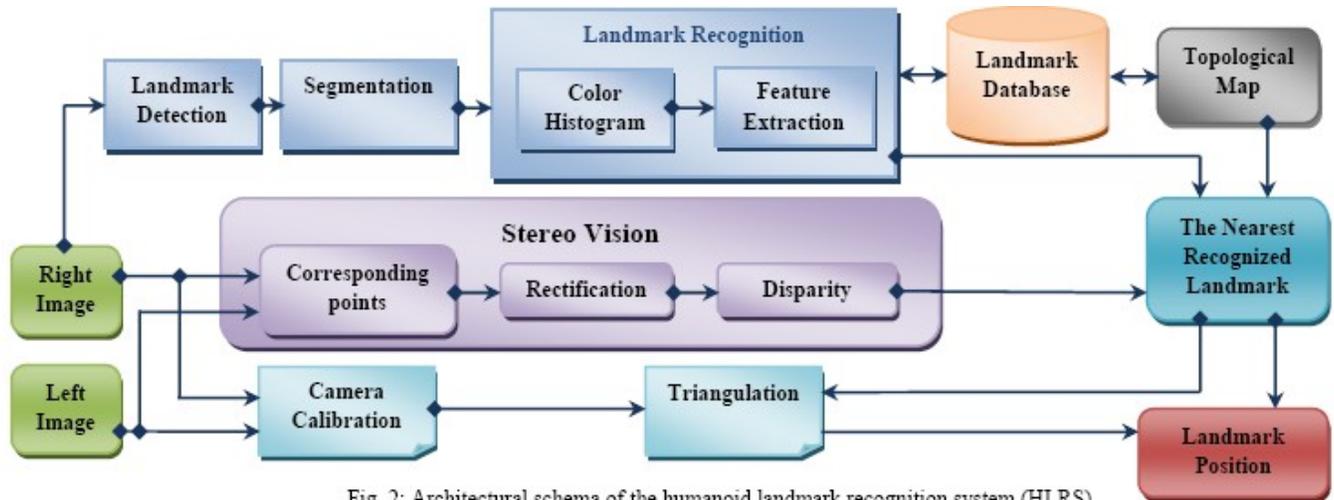


Fig. 2: Architectural schema of the humanoid landmark recognition system (HLRS)

Calibrating stereo cameras is implemented by calibrating each camera independently and then applying geometric transformation of the external parameters to find out the geometry of the stereo setting. The external camera parameters are needed for both the correspondence problem, which determines the epipolar lines for determining point correspondences, and for triangulation, which is used for reconstruction. We use the left camera as world reference system, so the parameters to be found are the translation vector and rotation matrix of the right camera with respect to the left one. We use the triangulation method to compute the world coordinate of all points in the images by using the disparity map, the focal distance of the two cameras and the geometry of the stereo setting (relative position and orientation of the cameras). The triangulation calculates the approximate position of the landmarks in the real world, which will be used during the path planning stage and also in executing the navigation task.

D. Stereo Vision and Disparity Map

Stereo vision has the advantage that it is able to obtain an accurate and detailed 3D representation of the environment around the robot by passive sensing and at a relatively low sensor cost. It is an important mechanism in robots, allowing judgments to be made based on the disparity between the images captured by each camera. We use humanoid stereo vision as a reliable and effective way to extract range information from the environment. The disparity map resulting from the stereo vision process is integrated with the landmark recognition stage to obtain the landmark that is detected nearest the robot.

The stereo vision process is divided into three steps. The first step is calculating the corresponding points in the left and right images. The left image is scanned to find corners with big eigenvalues and removes the features that are too close to stronger features. Therefore, we use the Lucas-Kanade optical flow in pyramids to calculate the coordinates of the feature points in the left captured frame and their corresponding points in the right captured frame. In the second step, the result is then rectified. It determines a

transformation of each image plane so that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes. The rectified images can be thought of as acquired by a new stereo rig, obtained by rotating the original cameras around the optical center. The important advantage of rectification is the computation of correspondences, by which a 2-D search problem is generally reduced to a 1-D search problem, typically along the horizontal raster lines of the rectified images [17]. The last step is the disparity map computation of the corresponding points in the left and right images, which is considered as the distance between them in pixels. We use the feature-based method by Birchfield et al. [18] to construct the disparity map. This method looks at features in one image and tries to find the corresponding feature in the other image. The features can be edges, lines, circles and curves. The main advantage of the feature-based algorithm is its speed. The process of finding features in both images, then calculating the disparity is carried out easily in real time.

E. Landmark Detection and Classification

Landmark classification is the core stage of our system. It is responsible for detecting and recognizing different landmarks during robot navigation. It consists of three main processes. First, the detection process which detects the landmarks in the captured image. Second, the segmentation process which segments the captured image into smaller images. These segmented images contain the detected landmarks which will be fed to the recognition process, the last component of landmark classification. The recognition process is used to classify the landmarks by using their color histogram to get an initial estimation of the landmarks. The SIFT classification is used to refine the recognition stage and obtains an accurate estimation of the landmarks. This combination of appearance-based and model-based methods leads to a robust classification of the landmark and also speeds up the classification process.

In our experiment, eight different landmarks were examined. These landmarks build models which are 63 cm high, 33 cm wide and 31 cm deep. Each landmark is a white model with a unique rectangular symbol attached at its top, as shown in fig. 3. These symbols are trademarks of supermarkets, restaurants, and department stores which will be processed to classify the landmarks in our miniature city. The implementations of the three processes of landmark classification are discussed in detail in the following paragraphs.



Fig. 3: Landmarks manipulated by the humanoid robot

Landmark detection is responsible for finding landmarks from the captured images. The purpose of this stage is to detect the landmarks' symbols. The detection stage is processed in four steps as shown in Fig. 4. The first step is the down and upscaling which is used to filter out the noise. It applies down and up sampling to the captured image by using Gaussian pyramid decomposition. In the second step, the canny filter is applied to finding the edges on the input image. The canny edge detector gives a good approximation of the optimal operator, i.e., the one that maximizes the product of signal-to-noise ratio and localization [19]. Third, dilating the canny filter output to remove potential holes between edge segments. The last step is to find and approximate the contours. We find all contours in the image and restrict it to the extreme outer contours, then approximate the contours by using the Douglas Peucker algorithm [20]. If the robot fails to detect any landmark in the captured image, it moves closer toward the landmark to capture the image again and repeats the detection stage.

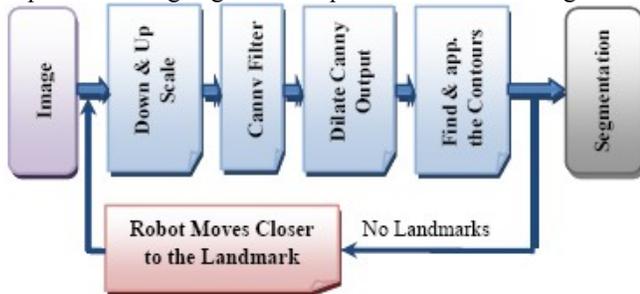


Fig. 4: Landmark detection steps

As an alternative to the computationally expensive windowing strategies, an image segmentation strategy is proposed. This method could improve results by reducing background clutter. We crop the detected landmarks from the image background to reduce the processing time during the recognition stage. The segmentation technique is based on the outputs of the detection clutter. The detection stage provides it with the proposed landmark regions, whereas

disparity provides it with the position of the landmarks with respect to the robot's position.

The recognition stage is implemented by using a two-step classification. The major advantages of the proposed two-step classification based method are its robustness and invariance towards scaling and translations. Also, it provides a good balance between fast processing time and high detection accuracy. First, we use the color histogram as an appearance-based method to get a fast rough classification of the landmark. A color histogram of the detected landmark is calculated first to produce initial hypotheses of the landmark which will be fed to the SIFT stage to get an accurate estimation of the landmark. It returns the hue distribution of the detected landmarks and does not preserve the geometric structures of these landmarks. The hue color component is determined by the dominant wavelength in the spectral distribution of light wavelengths. The hue component is ideally independent of the lighting conditions and the distance between object and observer. It is thus a reliable parameter for object recognition. On the other hand, color histograms of training images, which are stored in the database, are computed offline to reduce the consumed time, and the histogram of the tested landmark only needs to be calculated for the segmented image. Comparison of these two distributions (detected and stored landmarks) which are represented in the form of histograms is made on the basis of the correlation coefficient k for these distributions, which has the form:

$$k = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_i [(X_i - \bar{X})^2 (Y_i - \bar{Y})^2]^{1/2}} \quad (1)$$

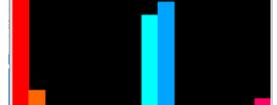
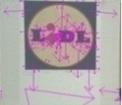
where X_i , Y_i are histograms for the considered distributions, \bar{X} , \bar{Y} are the average values of these distributions.

The largest correlated histograms are chosen as input to the SIFT processing step. After applying the color histogram method and obtaining some hypotheses, we calculate SIFT for the detected landmark and compare it with the stored SIFT values of the proposed landmarks from the database. The resulting hypotheses from the color histogram decrease the search time in the SIFT database. The landmark with the highest matching points is chosen as the recognized landmark. Tab. 1 shows the color histograms and the SIFT features of some detected landmarks. SIFT matching between these landmarks and the stored landmarks are shown in fig. 5. The average detection and recognition time is approximately 700 ms.

After recognizing the landmarks, the topological route map is used to specify which landmarks will be processed by the robot. The robot focuses only on the landmarks which are already mentioned in the route description and presented in the topological map. This leads to decreasing the processing time by ignoring unwanted landmarks. The nearest landmark in the route description to the robot is chosen by using the disparity map, and then triangulation is used to calculate the approximate world position of this

landmark. After recognizing landmarks and calculating their locations in the real world, the robot navigates to the landmark by using the information supplied by the topological map.

TABLE 1
COLOR HISTOGRAMS AND SIFT FEATURES OF SOME
LANDMARKS

Landmark	Color Histogram	SIFT
		
		
		
		

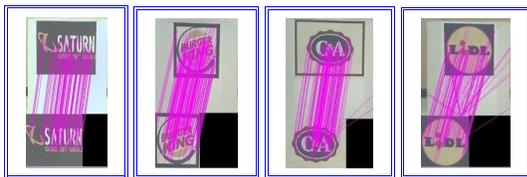


Fig 5: SIFT matching between the detected and the stored landmarks



Fig 6: SIFT matching under occlusions and different view angles.

VI. CONCLUSION

In this paper, we have presented our current effort toward building a robust and fast landmark recognition system. We used a more natural approach in terms of computational efficiency to recognize the landmark online during robot navigation. The color histograms of the detected landmarks are used to provide the rough initial estimate of the landmark. Then we processed the resulted hypotheses with SIFT to calculate an accurate estimation of the landmark.

The route topological map is used to decrease the processing time by processing only the landmarks mentioned in the route description and ignoring other landmarks during navigation.

In addition, we used the robot's stereo vision combined with the classified landmarks to find the nearest landmark to the robot and also to calculate the landmark's position in the real world.

REFERENCES

- [1] S. Ekvall, P. Jensfelt, and D. Kragic, "Integrating active mobile robot object recognition and slam in natural environments," in Proc. of the IEEE/RSJ International Conference on Robotics and Automation (IROS'06), Beijing, China, 2006.
- [2] A. Gopalakrishnan and A. Sekmen, "Vision-based mobile robot learning and navigation," *RO-MAN*, 2005.
- [3] A. Ramisa, S. Vasudevan, R. Lopez de Mantaras, and R. Siegwart, "A Tale of Two Object Recognition Methods for Mobile Robots". 6th International Conference on Computer Vision Systems, Santorini, Greece, May 12-15, 2008.
- [4] S. Ekvall, D. Kragic, and P. Jensfelt, "Object Detection and Mapping for Service Robot Tasks," *Robotica: International Journal of Information, Education and Research in Robotics and Artificial Intelligence*, vol.25, No. 2, pp 175-187, 2007.
- [5] Staffan Ekvall, Danica Kragic, and Frank Hoffmann, "Object recognition and pose estimation using color cooccurrence histograms and geometric modeling". *Image Vision Comput.* 23(11): 943-955, 2005.
- [6] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 60(2), 91-110, 2004.
- [7] Roland Siegwart, and R. Nourbakhsh, "Introduction to Autonomous Mobile Robots," Massachusetts, 2004.
- [8] R. Bianchi, A. Ramisa, R. Lopez de Mantaras, "Learning to select Object Recognition Methods for Autonomous Mobile Robots". 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008.
- [9] D. Lowe, "*Perceptual Organisation and Visual Recognition*. Robotics: Vision, Manipulation and Sensors," Dordrecht, NL: Kluwer Academic Publishers, ISBN 0-89838-172-X. 1985.
- [10] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer Vision and Pattern Recognition*, 2006 IEEE Computer Society Conference on, volume 2, pp. 2161-2168, 2006.
- [11] Jin-Seob Lee, Ji-Wook Kwon, Dongkyoung Chwa, and Suk-Kyo Hong, "Object detection of mobile robot using data-mining algorithm," *Control, Automation and Systems*, 2007. ICCAS '07. International Conference on, PP 1962-1965, 2007.
- [12] Juan Fasola, Manuela M. Veloso, "Real-time Object Detection using Segmented and Grayscale Images". *ICRA 2006*: 4088-4093, (2006).
- [13] B. Browning, and M. Veloso, "Real-time, adaptive color-based robot vision," *Intelligent Robots and Systems*, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on, 3871-3876, 2005.
- [14] M. Elmogy, C. Habel, and J. Zhang, "Robot Topological Map Generation from Formal Route Instructions," *Workshop Proceedings of the 2008 European Conference on Artificial Intelligence (ECAI)*, 6th International Cognitive Robotics Workshop (CogRob08), Patras, Greece, 2008.
- [15] Fujitsu Automation Co., Ltd. "HOAP-2 Instruction Manual," Third Edition, 2004.
- [16] S. Florczyk, "Robot Vision: Video-based Indoor Exploration with Autonomous and Mobile Robots," WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, ISBN: 3-527-40544-5, 2005.
- [17] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications*, 12(1):16-22, 2000.
- [18] S. Birchfield and C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo," *International Journal of Computer Vision*, 35(3): 269-293, December 1999.
- [19] M. Sonka, V. Hlavac, and R. Boyle, "Image Processing, Analysis, and Machine Vision," Thomson, 3th edition, 2008.
- [20] J. Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker," Intel Corporation, Microprocessor Research Labs 2000.