Landmark Manipulation System for Mobile Robot Navigation

Mohammed Elmogy
Dept. of Information Systems, Faculty of Computers &
Information Sciences, Mansoura University
Mansoura, Egypt
melmogy@mans.edu.eg

Abstract—**In mobile robot scenarios, it is expected that the robot autonomously navigates through home or office environments and processes objects/landmarks during navigation. Landmark manipulation is identified as one important research area in robot navigation systems. We have developed an online robot landmark processing system (RLPS) to detect, classify, and localize different types of landmarks during robot navigation. The RLPS is based on a two-step classification stage which is robust and invariant towards scaling and translations. It provides a good balance between fast processing time and high detection accuracy by combining the strengths of appearance-based and model-based object classification techniques. The experimental results showed that the RLPS is more powerful as it recognizes a wide range of landmarks and efficiently handles landmarks with occlusions, viewpoint variances, and illumination changes.**

Keywords-robot vision; object detection and classification; vision-based robot navigation.

I. INTRODUCTION

Vision is one of the most powerful and popular sensing method used for autonomous robot navigation that continues to demand a lot of attention from the mobile robot research community. When vision sensor compared with other on-board sensing techniques, the vision sensor has the ability to provide detailed information about the environment which may not be available using combinations of other types of sensors. The past decade has seen the rapid development of vision based sensing for indoor mobile robot navigation tasks.

Object manipulation in vision-based 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 [1]. For robot navigation, object recognition has the ability to learn and detect hundreds of arbitrary objects in images from uncontrolled environments which can be considered as a major breakthrough for many intelligent robotics applications.

On the other hand, object recognition in real scenes is deemed as one of the most challenging problems in computer vision. The visual appearance of objects can change enormously due to viewpoint variation, occlusions, illumination changes, or sensor noise. Furthermore, objects are not presented alone to the vision system, but they are immersed in an environment with other elements, which clutter the scene and make recognition more complicated. In a mobile robotics scenario, a new challenge is added to the last list: computational complexity [2]. In a dynamic world, information about the objects in the scene can become obsolete even before it is ready to be used if the recognition algorithm is not fast enough to handle the current robot's view.

This paper is organized as follows. Section II discusses some current related work. In section III, the architecture of our robot landmark manipulation system (RLMS), which is used to detect, classify, and localize landmarks for the mobile robot, is introduced. Section IV discusses the implementation and experimental results. Finally, the conclusion is presented in section V.

II. RELATED WORK

In some real-time robot applications, vision systems employing region segmentation by color to detect objects or landmarks during interaction with humans or navigating in a dynamic world. For example, Bruce et al. [3] implemented a segmentation system capable of tracking several hundred regions. It can classify each pixel in a full resolution captured color image. Michel et al. [4] build an occupancy grid from the synthesized top-down floor view, a step of color segmentation is performed in YUV space. They defined segmentation thresholds by sampling pixel values offline for obstacles placed in a variety of locations on the floor.

On the other hand, 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.

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 [5]. 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.

Recently significant work has been done in visual object classification, but few of them actually scale to the demands posed by mobile robot scenarios. In robotic applications, the robots should have lightweight, fast, and robust object perception methods that allow them to interact with the environment in real-time. For example, Lowe [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. This object recognition approach is 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.

Color-based object recognition, where objects of interest are colored in a uniquely identifiable and known way, is a technique that has found wide use in the robotics community. Therefore, there are many researchers who implemented their mobile robot vision applications by using color-based object recognition methods. For example, the authors in [7] proposed a recognition scheme that is based on the color co-occurrence histograms (CCHs). It is used in a classical learning framework that facilitates a "winner-takes-all" strategy across different scales. The detected "windows of attention" are compared with training images of the object for which the pose is known. The orientation of the object is estimated as the weighted average among competitive poses, in which the weight increases proportional to the degree of matching between the training and the segmented image histograms.

Fasola and Veloso [8] described an approach that performs visual object detection in real-time by combining the strength of processing the color segmented image along with that of the grayscale image of the same scene. They used color segmented images for producing initial hypotheses for the location of robots in the image, and grayscale images for final classification purposes.

III. System Architecture

We have developed an online robot landmark manipulation system (RLMS) running on a mobile robot. RLMS is used to detect, classify, and localize different types of landmarks during robot navigation. The robot autonomously navigates in an indoor environment, moves according to the processed route description, and localizes its position in the environment with respect to the detected landmarks. The robot recognizes predefined landmarks, estimates their position in the environment, and integrates the result with the localization module to automatically process the landmarks and use them in motion planning. Our main goal is to implement a robust, accurate, and real-time landmark manipulation system for mobile robot navigation which can handle different types of landmarks.

The robot uses the symbolic representation and the generated topological map of the route description to decide which landmark will be processed during navigation [9]. The topological map represents the route description in a graphical representation and it also retrieves the relationships between the landmarks to handle uncertainties during robot navigation. The symbolic representation of the route is grounded to the output of RLMS by using perceptual anchoring. The result is supplied to the motion planner to generate the shortest feasible path for the robot [10].

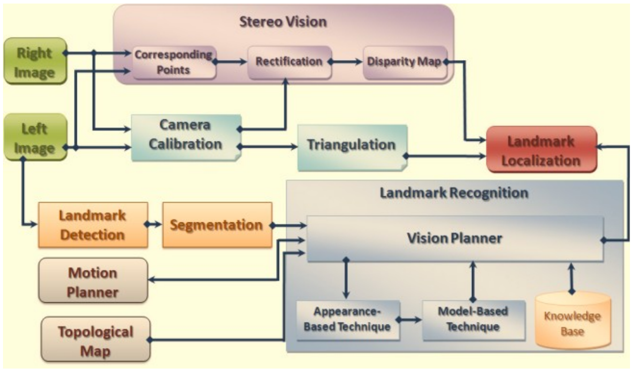

Figure 1. The architecture of the robot landmark manipulation system.

As shown in Fig. 1, the architecture of RLMS can be divided into three basic stages: stereo calibration, stereo vision and triangulation, and landmark classification. The stereo calibration is used to calculate the internal and external parameters of the left and right cameras of the robot. The stereo vision stage is responsible for creating disparity and depth maps of the captured images. The outputs from the stereo vision combined with the external parameters of the stereo pairs are used to calculate the 3D position of the retrieved landmarks in the real world. Last but not the least, the landmark classification stage is responsible for detecting and recognizing the landmarks from the captured frames. In the following subsections, the main building blocks of RLMS will be discussed in detail.

A. Stereo Vision and Landmark Localization

We use 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 classification stage to obtain the position of the nearest landmarks to the robot. The stereo vision process is divided into four main steps: corresponding calculation, rectification, disparity map generation, and triangulation.

In the corresponding point's calculation stage, the features points in the left image are retrieved and their equivalent features in the right image are determined. The left image is scanned to find corners with big eigenvalues and then the features that are too close to stronger features are removed. Therefore, we use the Lucas-Kanade optical flow in pyramids [11] to calculate the coordinates of the feature points in the left captured frame and their corresponding points in the right captured frame.

The rectification stage is used to determine a transformation of each image plane so that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes. Fig. 2 shows the resulting rectified image for a stereo. The average error of epipolar geometry, which is computed by all good matches, is 0.6155 pixel. Afterwards, the distance between the corresponding points in the left and right images in pixels is computed. The output of this step is a disparity map, where the disparities are the differences in x-coordinates on the image planes of the same feature viewed in the left and right cameras. We use the feature-based method by Birchfield and Tomasi [12] to construct the disparity map. The main advantage of the feature-based algorithm is its speed. The process of finding features in both images and then calculating the disparity is carried out easily in real time. Fig. 3 shows the disparity and depth maps of the stereo images illustrated in Fig. 2.

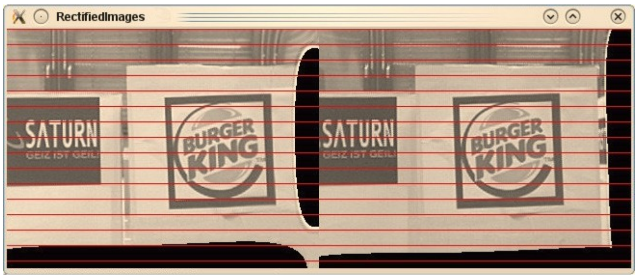

Figure 2. The resulting rectified image of the stereo pair.

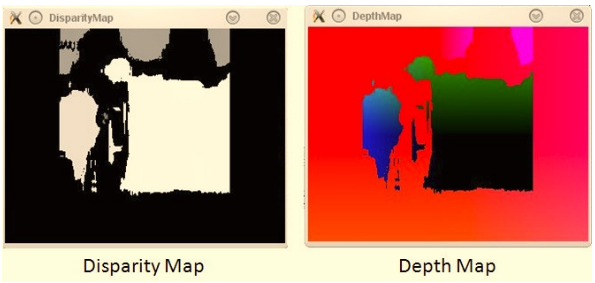

Figure 3. The disparity and depth maps of the stereo images.

Finally, to calculate the object's position in the 3D environment, the disparity map should be turned into distances by triangulation. This step is called reprojection, and the output is a depth map.

B. Landmark Detection and Segmentation

In robotic scenarios, the landmark detection stage should be capable of processing images extremely rapidly and of achieving high detection rates. The landmark detection stage is responsible for detecting and segmenting different types of landmarks from the captured image during robot navigation. Its main goal is to retrieve the landmarks from the captured image based on their shape and rejects false positives. It then segments the detected landmarks from the image background and stores them into individual images. These images will be fed to the recognition stage to classify the landmarks depending on both their features and the processed route. We used 9 different landmarks to examine our system as shown in Fig. 4. All data of these landmarks are stored in the knowledge base. These landmarks have logos of supermarkets, department stores, and restaurants; such as Lidl, C&A, and Burger King, respectively.

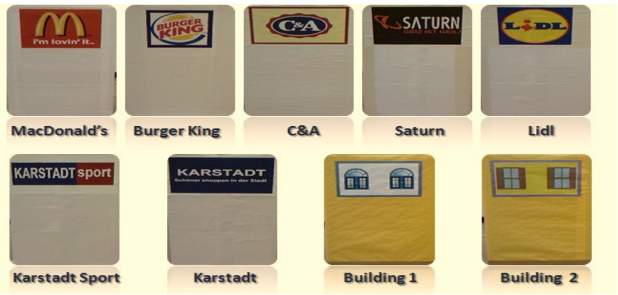

Figure 4. A set of landmarks used in our system.

The landmark detection stage is processed in four basic steps as shown in Fig. 5. The first step is the down- and up-scaling process 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 find the edges in the input image. The canny edge detector gives a good approximation of the optimal operator. It maximizes the product of signal-to-noise ratio and localization [11]. The third step is to dilate the canny filter output to remove potential holes between edge segments. Therefore, the image is then dilated to connect any small areas that may be disconnected despite the large hysteresis in the Canny filter [11]. 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 [13].

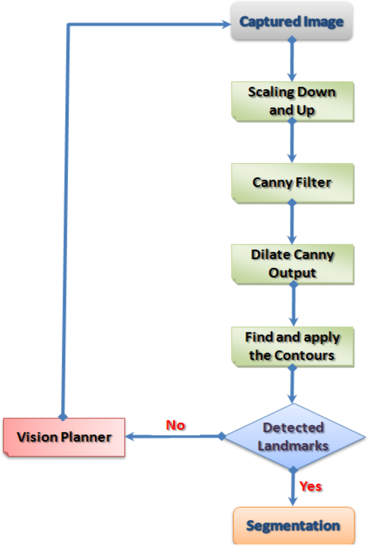

Figure 5. The flow diagram of the landmark detection stage.

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. This allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. 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. Once the object is segmented from the background, it has to be represented in a compact way for future indexing.

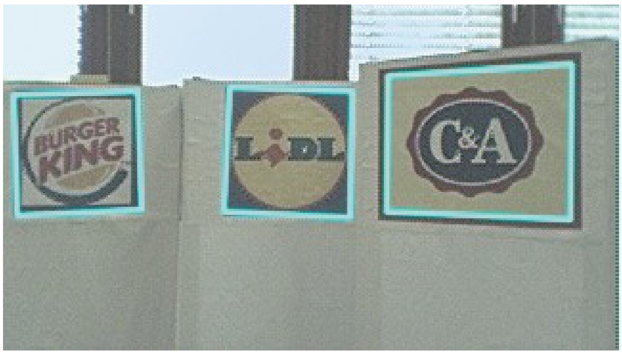

Figure 6. The output of the detection stage.

C. *Vision Planner*

The vision planner decides which algorithm should be used to recognize landmarks seen by the robot during navigation. This learning is based on both simple attributes extracted on-line from the images and data retrieved from the symbolic and graphical representations of the processed route. For the appearance-based approaches, the vision planner can choose between Hough transform to detect street boundaries and crossroads, color histogram to detect landmarks with logo (see Fig. 4), or color detection to detect other buildings in our miniature city. For the model-based approaches, it can choose whether or not to use the SIFT approach. In this paper, we focus only the logo landmarks.

Finally, the vision planner stores the recognized landmarks and their positions. It can use this information to get an idea of where the position of the new landmarks is with respect to the detected landmarks. The vision planner provides the vision output to the symbol grounding stage in the motion planner to connect the symbolic representation of the landmarks with their equivalent physical data.

D. *Landmark Classification*

Landmark classification is the core stage of RLMS. It 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, an appearance-based method is used to classify the landmarks to get an initial estimation of the processed landmark. Then, a model-based classification is used to refine the recognition stage and obtain an accurate estimation of the landmark. This combination of appearance-based and model-based methods leads to a robust classification of the landmark and also speeds up the recognition process.

We use the color histogram as appearance-based method to get a fast rough classification of the landmarks. Selecting which algorithm should be used by a mobile robot computer vision system is a decision that is usually made a priori to the processing stage of the captured image. A color histogram is calculated first to produce initial hypotheses before supplying the processed landmark to the model-based stage to get an accurate estimation of this landmark. As the RGB color space is not very stable with regard to alterations in the illumination, the representation of a color with the RGB color space contains no separation between the illumination and the color parts. Therefore, we used the HSV color space, which is robust against alterations in illumination, because the color parts and the illumination are represented separately. The color histogram 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. This component is ideally independent of the lighting conditions and the distance between object and observer. Therefore, they are reliable parameters 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. Only the histogram of the tested landmark needs to be calculated. 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 for these distributions. If the two histograms are identical, the correlation coefficient is equal to unity. The stronger the correlation coefficient differs from unity, the stronger is the diversity between the considered distributions. Thus the correlation method of comparison of histograms is the simplest and strongest method for the analysis of observed data for a certain meteorological phenomenon. The resulting correlation coefficients of the color histograms with the information retrieved from the topological map and the route symbolic representation are used as an initial estimation of the processed landmark. Fig. 7 shows the hue color histograms of the tested landmarks.

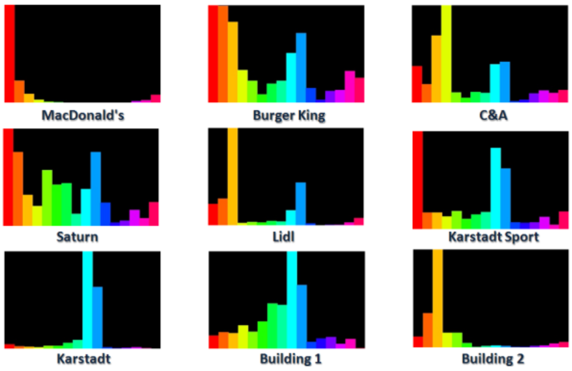

Figure 7. The color histograms of the tested landmarks.

Afterwards, the SIFT technique is used to classify the landmarks according to their geometrical properties. After applying the appearance-based technique and having some hypotheses, we used the model-based stage to refine the recognition stage and obtain an accurate estimation of the landmark. Initially, SIFT descriptors of the detected features in the processed landmark are determined. Then the resulting descriptors are matched to the ones with the highest hypotheses which are stored in the landmark knowledge base by using the Euclidean distance. False matches are rejected if the distance of the first nearest neighbor is not distinctive enough when compared with that of the second.

Once a set of matches is found, the Hough Transform is used to cluster each match of every knowledge base image depending on its particular transformation. Although imprecise, this step generates a number of initial coherent object hypotheses and removes a notable portion of the outliers that could potentially confuse more precise but also more sensitive methods. All clusters with at least three matches for a particular training object are accepted, and fed to the next stage: the Least Squares method, used to improve the estimation of the affine transformation between the model and the test images. The matching process between the examined images and the stored landmarks is shown in Fig. 8.

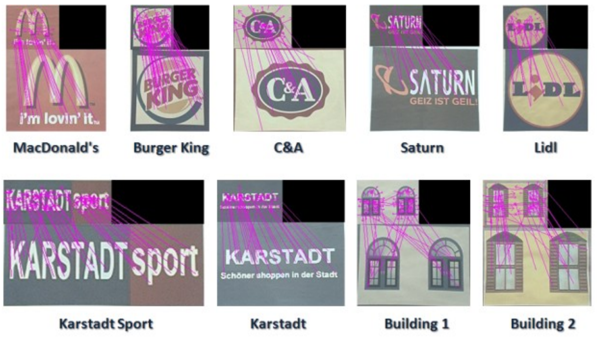

Figure 8. The matching between the calculated SIFT descriptors of the processed images and those of the stored landmarks in the knowledge base.

Finally, the landmark with the highest matching points is chosen as the recognized landmark (i.e. the more SIFT-matches found in an image, the more likely it is that that image contains the object). If the number of matches exceeds an object-dependent threshold, the object is considered recognized. Some objects have more features than others and are thus easier to recognize. To minimize the number of false positives, the threshold depends on the number of features found during training.

After recognizing the landmarks, the topological 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 be 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.

IV. Implementation and Experimental results

In this section, we demonstrate that RLMS is more powerful as it learned to recognize a wide range of landmarks and find them in the environment even when there are landmarks of similar colors in the field of view. It also handles landmarks with viewpoint variances, occlusions, and illumination changes. The main purpose of this experiment is to evaluate our proposed technique to be used in an indoor mobile robot. We implemented our system by using two 0.25" CMOS unsynchronized cameras. The two cameras are linked via USB connections to an Intel 1.73GHZ Duo laptop with 3GB RAM. The operating system is a standard Linux distribution and kernel with Video for Linux drivers for video capture. In its current form the system can process 320x240 images at 25fps.

TABLE I. The resulting internal and external calibration parameters of the robot's cameras.

Parameters	Values

Left intrinsic matrix	457.5843 000.0000 153.1480
	000.0000 457.5843 135.7438
	000.0000 000.0000 001.0000

Left distortion matrix	0.4296 3.2729 0.0000 0.0000 -35.5685

Right intrinsic matrix	457.5843 000.0000 172.7118
	000.0000 457.5843 148.4658
	000.0000 000.0000 001.0000

Right distortion matrix	0.4763 1.2530 0.0000 0.0000 -21.4564

Stereo rotation matrix	0.9998 -0.0068 -0.0191
	0.0060 0.9992 -0.0398
	0.0194 0.0397 0.9990

Stereo translation matrix	-2.5325 -0.0634 0.0187

To evaluate the performance of our robot landmark manipulation system, the following procedure was carried out: first, we approximately captured 900 different images for the tested landmarks with different viewing angles, distances from the robot, occlusions, and illumination changes. Each tested image contains one or more instances of the landmarks, some of them with illumination changes, partial occlusions, or viewpoint variances. In addition to our proposed method, we selected three state-of-the-art object recognition methods, which are suitable to be adapted to the mobile robot's applications, to compare them with the performance of our proposed technique. We used the original SIFT, color histogram, and Speeded Up Robust Features (SURF) [14] techniques to compare them with the detection rate of the proposed system. Evaluation is done by comparing the number of detected objects, false negatives, and false positives resulting by applying these approaches to the processed dataset.

The ratio of the correct classifications for each tested landmark in the dataset is illustrated in Fig. 9. It can be observed that the correct recognition rate of our proposed technique is over that 85% for the most of the tested landmarks. It also can be noticed that the detection rate of the color histogram is the lowest rate among the other tested techniques. For SIFT and SURF methods, the landmark classification is based on the more matches found in the image, the more likely it contains the landmark. Table II illustrates the detection rate for each technique. As seen in the table, our proposed system was able to achieve 91% classification accuracy on the test set. On the other hand, it can be noticed that our system handles the cropped landmarks efficiently except in a few cases which result from false positives in the first recognition stage.

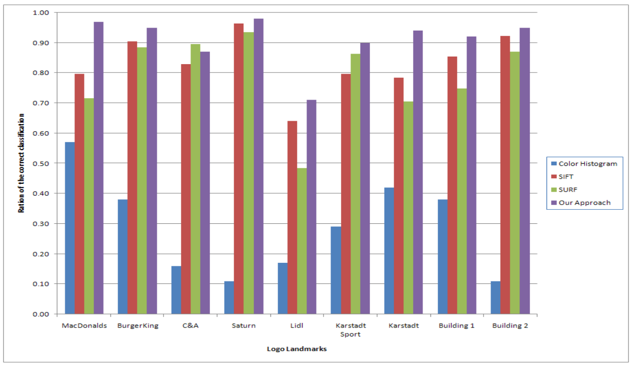

Figure 9. The ratio of the correct classifications with the whole dataset of the processed images.

TABLE II. DETECTION RATE OF THE TESTED TECHNIQUES.

Correct classification of	Color Histogram	SIFT	SURF	Our Approach
All tested images	42.22%	80.74%	84.06%	90.74%
Cropped images	12.70%	96.83%	97.09%	94.44%
Different viewing angle images	58.33%	72.69%	74.42%	88.19%

For each tested classification methods, we measured the average time for detection, average classification time, and average total manipulation time. Table III lists the time consumed in each stage.

TABLE III. LANDMARK MANIPULATION TIME

Time (msec)	Color Histogram	SIFT	SURF	Our Approach
Detection	104.5	-----	-----	104.5
Appearance-based Classification	3.4	-----	-----	3.4
Feature Extraction	-----	549.3	186.9	549.3
Finiding Correspondings	-----	320.2	170.6	320.2
Total Average Manipulation Time	107.9	3431.4	2470.0	1137.6

V. CONCLUSION

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

Stereo vision is used to calculate the 3D position of the landmarks in the real world. We used 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 classification stage to obtain the position of the landmarks nearest to the robot.

REFERENCES

[1] A. Gopalakrishnan, S. Greene, and A. Sekmen, “Vision-based mobile robot learning and navigation,” In IEEE International Workshop on Robot and Human Interactive Communication (ROMAN 2005), 2005.

[2] A. Ramisa, S. Vasudevan, D. Aldavert, R. Toledo, and R. Lopez de Mantaras, “Evaluation of the sift object recognition method in mobile robots,” In 12th International Conference of the ACIA. Frontiers in Artificial Intelligence and Applications, vol. 202, pp. 9–18, 2009.

[3] J. Bruce, T. Balch, and M. Veloso, “Fast and inexpensive color image segmentation for interactive robots,” In proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), vol. 3, pp. 2061–2066, Octobar 2000.

[4] P. Michel, J. Chestnutt, S. Kagami, K. Nishiwaki, J. J. Kuffner, and T. Kanade, “Online environment reconstruction for biped navigation,” In proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA’06), pp. 3089– 3094, 2006.

[5] P. M. Roth and M. Winter, “Survey of appearance-based methods for object recognition,” Technical report, Institute for Computer Graphics and Vision, Graz University of Technology, Austria, 2008.

[6] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, no. 60, vol. 2, pp. 91–110, 2004.

[7] S. Ekvall, F. Hoffmann, and D. Kragic, “Object recognition and pose estimation for robotic manipulation using color cooccurrence histograms,” In proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03), vol. 2, pp. 1284–1289, 2003.

[8] J. Fasola and M. Veloso, “Real-time object detection using segmented and grayscale images,” In proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA'06), pp. 4088–4093, May 2006.

[9] M. Elmogy, C. Habel, and J.Zhang, “A Cognitively Motivated Route-Interface for Mobile Robot Navigation,” Cognitive Systems Monographs, vol.6 , pp.73–82. Springer Berlin/Heidelberg, 2009.

[10] M. Elmogy, C. Habel, and J. Zhang, “Time efficient hybrid motion planning algorithm for hoap-2 humanoid robot,” In ISR/ROBOTIK 2010, pp. 1046–1053, Munich, Germany, 2010.

[11] G. Bradski and A. Kaebler, "Learning OpenCV", 1st ed. O’Reilly, 2008.

[12] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-topixel stereo". International Journal of Computer Vision vol.35, no.3 (1999), 269–293.

[13] M. Sonka, V. Halvac, and R. Boyle, “Image Processing Analysis and computer vision,” Thomson, 3ed., 2007.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. Van Goo, “Speeded-up robust features (SURF),” Computer Vision and Image Understanding (CVIU), no. 110, vol. 3, pp. 346–359, 2008.

[15] R. Bianchi, A. Ramisa, and R. Lpez de Mntaras, “Learning to select object recognition methods for autonomous mobile robots,” In 18th European Conference on Artificial Intelligence, pp. 927–928, Patras, Greece, July 2008.

[16] S. Lazebnik, C. Schmid, and J.Ponce, “Beyond bag-offeatures: Spatial pyramid matching for recognizing natural scene categories,” In Computer Vision and Pattern Recognition, IEEE Computer Society Conference(CVPR’06), vol. 2, pp. 2169–2178, 2006.