Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Object Recognition from Omnidirectional Visual Sensing for Mobile Robot Applications

Min-Liang Wang and Huei-Yung Lin
Department of Electrical Engineering
National Chung Cheng University
Chiayi 621, Taiwan
poolllz.cgsp@msa.hinet.net,hylin@ccu.edu.tw

***Abstract*—This paper presents a practical optimization procedure for object detection and recognition algorithms. It is suitable for object recognition using a catadioptric omnidirectional vision system mounted on a mobile robot. We use the SIFT descriptor to obtain image features of the objects and the environment. First, sample object images are given for training and optimization procedures. Bayesian classification is used to train various test objects based on different SIFT vectors. The system selects the features based on the k-means group to predict the possible object from the candidate regions of the images. It is thus able to detect the object with arbitrary shape without the 3D information. The feature optimization procedure makes the object features more stable for recognition and classification. Experimental results are presented for real scene images captured by a catadioptric omni-vision camera.**

I. Introduction

Summarizing the object recognition task that asks "what is the object in the image?" actually we mean that "does it corresponds to any model that we know?" If not, the object is new to our database. Otherwise, it is recognized as an existing object. For human beings, the database corresponds to the memory in our brain. But for a robot, there is usually no prior knowledge of the objects for recognition. Thus, it is mandatory to establish the object models prior to the recognition tasks.

For our purposes, object recognition consists of two basic steps– identifying and localizing the object. Identification determines the nature of the imaged objects. For instance, we may want to know whether there is a ball among the objects in an image, or whether the only object we are looking at is indeed a ball. Localization determines the position of the object in a view. In this work, we use a mobile robot to explore the environment for object identification (see Fig. 1).

A real-world environment is usually highly complicated, uncertain, and contains many scattered features. We use a catadioptric omnidirectional camera to find stable and interested features, which can be recognized from a nearby robot. In this paper, multiple rules are adopted. Firstly, the object is separated from its background, similar to the work presented in [1]. Secondly, we use active vision to find the stable features. Some researchers such as Ballard [2] also uses similar approach to simplify visual perception. Finally, we use active vision to gather more information for object recognition. This is particularly important in the real-world ambiguous situations.

In addition to the use of active vision, we propose a method to optimize the number of keypoints in the object database as well as the environment images. Like the previous approaches on object recognition, our model illustrates the objects by a set of interested features with spatial proximities [3], [4], [5].

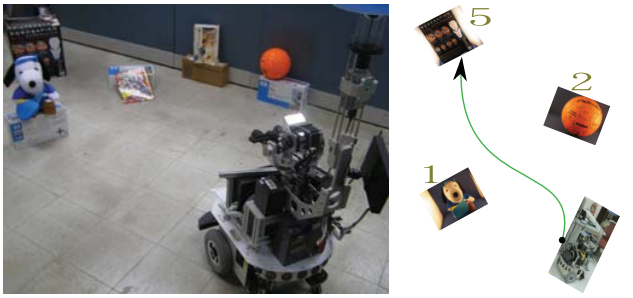

Fig. 1. The experimental setup for object recognition using our mobile robot. The robot has the knowledge of several object models, which is used for object identification and localization in an unknown environment.

Object recognition is still an open problem in computer vision, and the reasons for this are numerous. Its applications to robotics are very popular and evolve rapidly. Thus, it has attracted a number of robotics researchers to deal with the related issues. The first question is how to extract the high level features. The second question is how to reduce the features to keep the object characteristics more stable. The aim of this work is to deal with the object recognition problem for mobile robot applications. Towards this aim, the problem can be stated as follows. Given:

1) The list of feature descriptors from a given classification model.
2) The list of feature descriptors detected in a real-world scene.
3) A list of constraints that model features must satisfy.

Find a mapping between the object model features and the environment image features such that the constraints satisfied by the model features are satisfied by the corresponding image features. The major contribution of this work is to improve the stability of feature detection algorithms in the real-world environment.

II. Previous Work

In this paper, we use the keypoints to refer to the image features detected by SIFT (Scale Invariant Feature Transform) descriptor [6]. Image features have been successfully used

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

for object recognition purposes [7], [8], [9], [10]. One of the reasons that SIFT is very successful in object recognition is that it uses a large number of points to represent the object. This makes the system bear to noise, and solve the problem of occlusions. However, there is a major drawback. Because the matching stage needs a lot of CPU time to compute the vector correspondences, a significant amount of computation in the recognition process is focused on matching the observed features with the object model feature database.

When performing an object recognition task, many of the acquired features look very similar. There are several reasons for this. First of all, these are the features corresponding to the same feature point on the object seen from different viewpoints. Secondly, there are similar features on repetitive structures of the same object. Finally, there might be ambiguous features on different objects. Some researchers have previously considered performing object recognition on mobile robots. Kragic and Bjorkman [11] adopt a combination of foveal and peripheral vision that uses structure from stereo and bottom-up visual saliency to identify objects. Gould et al. [12] construct a vision system which does perform targeted object detection. Therefore, the vision system highlights tracking of previously recognized objects and depends on reasonably reliable recognition in its surrounding view. Ekvall et al. [13] propose a system utilizing the top-down information to lead their visual system by training a resiliency-like visual map to shoot some objects. Their object recognition system is also based on SIFT features.

III. System Overview

In this paper, we use a feature based method to extract the characteristics of an object, and make the characteristics more stable. Here we assume that the object images are given. Therefore, the object database must contain the features and the classification parameters have to be trained from the object images in advance. The first problem is to select a feature detection method. Secondly, the features must have sufficient characteristics to represent the object.

A. Feature Detection and Clustering

We focus on feature detection and its stability in this section. As described in the introduction, the object model has to be constructed before the object detection task. Because the catadioptric omni-vision system is used, there are two problems related to the feature detection. One is that there are too many features in the real scene image if we use all of the features for classification and generate the constraint (object model). There contain too many similar features to represent different object models– a feature might be selected by several objects. This might have an impact on the uncertainness of object classification in both the training and detection steps. The other problem is that the image is distorted when captured from the catadioptric omni-vision system.

To solve these two problems, we select a stable feature detection method and optimize the object features to obtain the independent features. In our implementation, the SIFT descriptor is used to detect the image features. Based on the detection results, the features are optimized to represent an object model.

Fig. 2. The object image (left figure) and the features (right figure).

For mobile robot applications, the environment is usually complicated for object searching and robot navigation. The image features are scattered all over the places in a real-world scene. Even a single object as shown in Fig. 2 has many detected features. There are a lot of features can be detected using SIFT feature detection, but some features are located at trivial positions. For this problem we use the k-means algorithm to roughly separate the SIFT features to several subgroups [14]. We use the cluster algorithm to cluster the SIFT feature set into k subgroups. The derived centroids become internal nodes. Recursively, the clusters are subdivided in the same manner until they consist of less than k elements. We then use our feature optimization method to optimize the features based on the subgroups [15], [16]. The feature optimization procedure is discussed in the section IV.

B. The Mobile Robot Platform

The mobile robot platform shown in Fig. 3 was used in this work. The robot is equipped with several sensors including a catadioptric omni-vision system, encoders, a stand-alone computer and sonars.

In this paper we focus on object recognition with only the omni-vision system. As mentioned in section III-A, the features might locate at some trivial positions. This phenomenon is caused by the catadioptric omni-vision system, which captures the visual information of the environment around the camera axis in one image on 360 degrees of circumference. We use SIFT to detect the features in the environment, and the features will distribute any place of the image. As mentioned above, we need to separate the features to several subgroups. Make some groups concentrate on the objects.

In order to easily process the omnidirectional image captured from the omni-vision system, we transform the omnidirectional image to a rectangular, panoramic image through a coordinate transformation. It converts the polar coordinates to the rectangle coordinates, which is also called "image warping" as shown in Fig. 4.

IV. Characterization and Simplification of the Object Recognition

Object recognition using the SIFT descriptor has a main drawback that the processing time needed increases with the

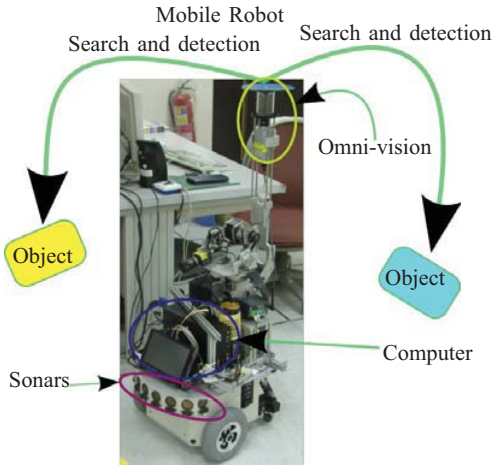

Fig. 3. The mobile robot platform used in the experiments. The encoders and the catadioptric omni-vision system mounted on the top of the robot are used for object searching.

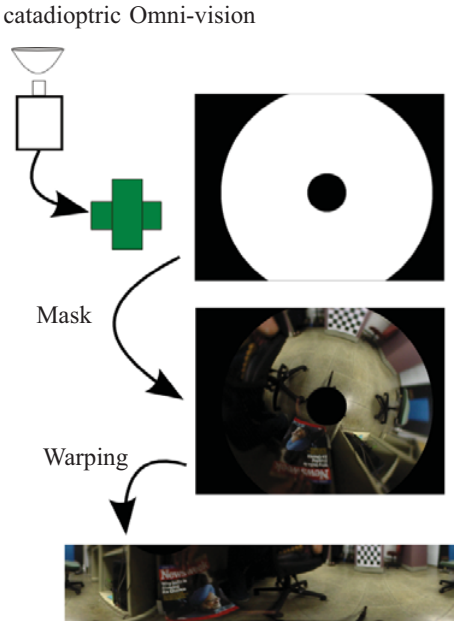

Fig. 4. The image warping technique used in this work.

number of features stored in the database. We therefore use a feature optimization procedure to enhance and reduce the cluster features corresponding to the object. In the object model training stage, we need to find the stable features of the object first. In this work we use the SIFT descriptor for the pure object image. Suppose we have stable SIFT features of an object, we can then adopt the Bayesian classification to train the object model.

First, suppose the prior feature vector probability distributions of an object is a normal distribution and can be represented as follow:

$$p(x|\mu) \sim N(\mu, \Sigma) \tag{1}$$

TABLE I
AN EXAMPLE OF THE FEATURE MATRIX, WHERE AN OBJECT CONTAINS n FEATURES AND EACH FEATURE HAS 128 DIMENSIONS.

Features	Object 1	$\cdots$	Object m
1	10 19 7 $\cdots$ 9	64 9 12 $\cdots$ 1	4 5 2 $\cdots$ 14
2	$\vdots$	$\cdots$	$\vdots$
$\vdots$	$\vdots$	$\cdots$	$\vdots$
n	5 12 3 $\cdots$ 4	$\cdots$	6 18 3 $\cdots$ 8
mean	9	$\cdots$	6
variance	0.21	$\cdots$	0.37

where μ and $\sum$ are the mean and variance of the SIFT features, respectively. In other words, we assume every object is represented by an normal distribution but with different mean and variance. Secondly, we assume there are stable SIFT features and each SIFT feature has 128 dimensions. Third, we use all features to derive a set of feature vectors from an object image. In other words, a set represents all the feature's vectors of one object image. Fourth, we use a matrix to represent all sets of objects. The form of the matrix is shown in Table I.

Our goal is to find the object model using discriminant functions and distinguish the objects through Bayesian classification. The Bayesian classification equations are given as follows:

$$P(w_i|x) = \frac{p(x|w_i)P(w_i)}{p(x)} \tag{2}$$

$$p(x) = \sum_{i=1}^{c} p(x|w_i)P(w_i) \tag{3}$$

where $p(x|w_i)$ is the likelihood of w_i with respect to x, and c is the the total number of objects that we know. As mentioned before we assume that each object has a Gaussian distribution, so $p(x|w_i)$ must be a Gaussian. The Gaussian parameters are calculated using the matrix obtained from all sets of feature vectors (see Table I). Therefore, we get the likelihood functions of all objects.

It is easily to calculate the distribution of an object. For example, suppose an object has n features. We calculate the vector mean of all features which belong to the object. The calculation of variance is carried out similarly. Refer to [17] for more details. Finally, different objects are distinguished by the discriminant function given as follow:

$$g_i(x) = \ln p(x|w_i) + \ln P(w_i) \tag{4}$$

In this section we assume that we have stable SIFT features to calculate the object model. In the next section, there is a number of procedures to make the SIFT features more stable without loss of the characterization of the object.

V. Optimal Feature Transform

A. Outlier Rejection

The characterization of an object is not enough if we use all of the features detected from an object to train the discernment

function. Suppose the SIFT features detected from an object image have the form:

$$D_{training} = \{v_1, v_2, \cdots, v_n\} \tag{5}$$

where $D_{training}$ is a set of SIFT features, v_i represents a feature point. We separate the features to several groups through k-means cluster. This can be computed as follows.

$$\begin{cases} k(D_{training}) & = \{k_1, k_2, \cdots, k_n\} \\ k_i & = \begin{Bmatrix} c_i \\ r_i \end{Bmatrix} \end{cases} \tag{6}$$

where each k_i is a group of features, and c_i is the center of the cluster, r_i is the maximum radius of the cluster, which cover all the features which belong to this cluster.

Suppose a group has n features. The following equation is minimized to make the cluster stable:

$$J(v, k) = \sum_{i=1}^{n} |v_i - k_i|^2 \tag{7}$$

where $J(v, k)$ is the target function, feature v_i belongs to the cluster k_i.

For the definition of outliers features, we hypothesize a feature which belongs to a cluster k_i and calculate the distance between the feature position to the k_i cluster center. We also calculate the other distances $D_{between\ distance}$ between the k_i cluster and the nearest cluster $k_{nearest}$, for $k_{nearest} \neq k_i$. If the distance is less than the maximum radius of cluster $k_{nearest}$, we define this feature is as an outlier. This can be stated as follow:

$$D_{between\ distance} = \|c_i - c_{nearest}\|^2 \tag{8}$$

where c_i is the distance between the feature position and its cluster owner. $c_{nearest}$ is the distance between the feature's cluster owner center and the nearest cluster center. Equation (8) is to calculate the distance between a cluster and its nearest cluster. In order to reject the outlier features, we modify the radius of each group radius in advance. We assume the maximum radius cannot cover the features of other group. The new radius of the cluster can be expressed as:

$$R_{new} = \frac{R_{old} \times D_{between\ distance}}{R_{old} + R_{nearest}} \tag{9}$$

where $R_{nearest}$ is the maximum radius of the nearest group. R_{old} is the maximum radius of this group. $D_{between\ distance}$ is calculated from equation (8). We remove the outlier features of each group, the constraint is:

$$\begin{cases} \|v_i - c_i\| < r_{new}, & \text{inlier} \\ \|v_i - c_i\| > r_{new}, & \text{outlier} \end{cases} \tag{10}$$

The procedure is shown in Fig. 5 and the pseudo code of this procedure is given in Section V.-C. Some features of the object might still not be stable, for the features located at shiny or dark positions in an image. In order to delete this kind of features, we select the features which are on a suitable region of the histogram.

B. Brightness Histogram Constrains

As mentioned before, we need to limit the features on a suitable region using the histogram constraint. Suppose I is a histogram of the image,

$$I = \frac{1}{q} \sum_{j=0}^{q-1} f(x, y) \tag{11}$$

where $f(x, y)$ is the gray value of a image position (x, y) and q is the maximum gray value. Then the feature v_i which maps to this region is a reliable feature.

C. Pseudo-code of the Optimal Feature Transform

To obtain the optimal features from input SIFT features, the optimal feature transform steps are as follows:

1) Separate the features to each group.
2) Reduce the features using the radius parameter.
3) Histogram sampling (select the middle region).
4) Save the features and feature vectors.

Algorithm 1 The pseudo-code of optimal feature transform

```
cluster all SIFT features use k-means
k-means()← allfeaures(f_i)
//separate the features to each group
for f_i = 0 to f_i = max do
    if f_i → cluster k_i then
        cluster k_i += 1
    end if
    //calculate each group's maximum radius
    if distance f_i is maximum distance then
        Maximu distancek_i ← f_i's distance
    end if
end for//calculate the new radius of each group
for k_i = 0 to k_i = max do
    new radius k_i ← {radius k_i + radius k_nearest}/2
end for//kill the outliers
for f_i = 0 to f_i = max do
    if distance f_i ≥new radius k_i then
        kill f_i
    end if
end for//brightness histogram constrains
for f_i = 0 to f_i = max do
    gray value f_i → g_i
    if dark region ≤ g_i or g_i ≥shiny region then
        kill f_i
    end if
end for//save remaining features
matrix ← features, vectors
```

VI. Procedure of the Proposed System

Summarizing the object recognition for a mobile robot, it is realized by the following steps:

1) Input the object images.
2) Derive the optimal feature transforms of object images.
3) Train and generate the classification parameter.

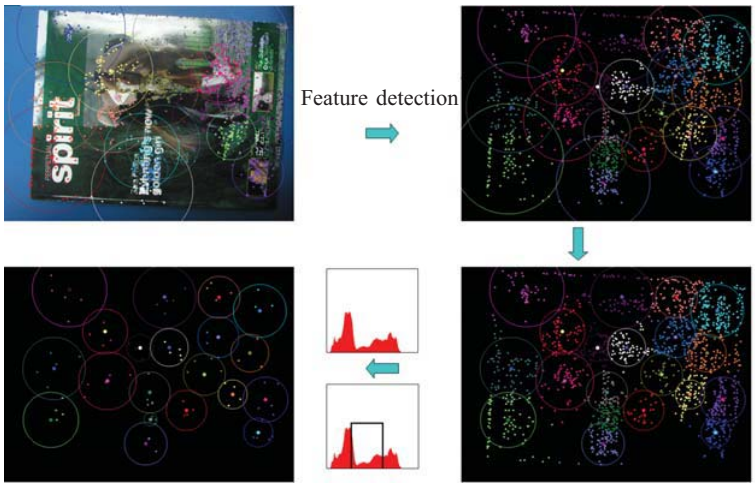

Fig. 5. We utilize the SIFT to detect the features and employ the k-means cluster. The top-left image is an object image with 2003 SIFT features. In the top-right image, only the features and the clusters are shown($k = 20$). The bottom-right image is with the modified radius. The bottom-left image shows the reliable regions and choosing 89 stable features.

4) Capture the environment image from catadioptric omni-vision camera.
5) Derive the optimal feature transforms of the images.
6) Select one of the objects which we want to search for.
7) Perform object detection and recognition.

Fig. 6 gives a flowchart of the our approach.

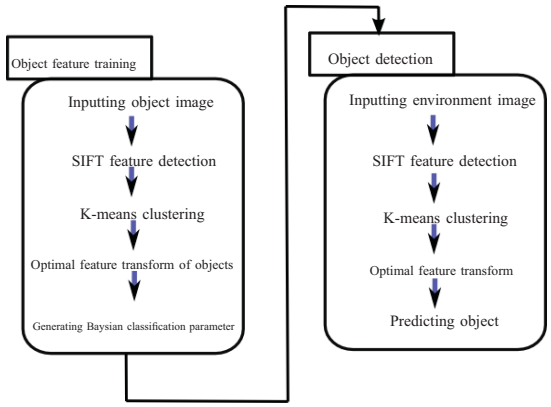

Fig. 6. Flowchart of the proposed system.

VII. Experiments

This paper used a catadioptric omni-vision system mounted on a mobile robot to search the object in an unknown environment. First, we trained and optimized the sample image of the objects. In the experiment, the robot navigated in a complex environment (as shown in Fig. 1). We selected an object which we would like to search for. If the environment map is known, then the object can be localized using the robot navigation information. The object models are shown in Fig. 9. Fig. 10 shows an example of search results.

Fig. 7. The object search result, the lines illustrate the matching between the object sample and the environment images.

TABLE II
Performance analysis. Due to the table size we only show results of object 2 (object number refer to Fig. 9). The other objects are also work using our method.

Object 2	Full features	Optimal features
Image size	1632 × 326	1632 × 326
Total features	2003	89
Average features matching error	22.7%	11.1%
Accuracy rate	40%	65%

The above experiment uses 6 books as the test data set to illustrate our object recognition algorithm. It is fairly difficult to recognize these similar objects in an unknown environment.

The feature matching errors are reduced as shown in Fig. 8. The analysis (see Table II) was carried out to illustrate that the optimization step increased the accuracy rate compared to the non-optimization features. In the experiments, object image may be small when the mobile robot is far away from the object. This may decrease the object features and cause the low accuracy results.

The processing time of the object search algorithm is about 2 second per frame from capturing an environment image to finding the object. More results can be found in http://vision.ee.ccu.edu.tw/poolllz/Projects.html.

(a) object model database 1.

(b) object model database 2.

Fig. 9. Each object has 2 or 3 images for the experiments. In our system, the omni-vision system is installed higher than the ground plane. One could think the object model is as a planar image on the ground due to the far distance between the omni-vision system and the object.

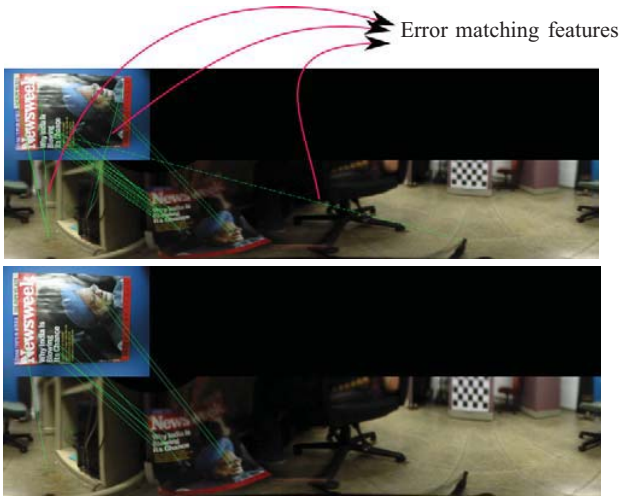

Fig. 8. The top image shows the SIFT feature matching without feature optimization. The bottom image shows the feature matching with the feature optimization. The comparison indicates the feature matching error decreased by using the feature optimization procedure.

Fig. 10. The object search result.

VIII. Conclusions and Future Work

We have proposed an optimal feature transform for object recognition with a mobile robot in an unknown environment. The results demonstrate that the accuracy rate increases with the optimal feature transform method. In the future work, SVM will be implemented instead of Bayesian classification. For real-time applications, the GPU based SIFT method [18] is a best manner to decrease the CPU computation time, which will also be considered.

Acknowledgment

The support of this work in part by the National Science Council of Taiwan, R.O.C., under Grant NSC-96-2221-E-194-016-MY2 is gratefully acknowledged.

References

[1] P. Fitzpatrick, "First contact: an active vision approach to segmentation," *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, pp. 2161–2166 vol.3, Oct. 2003.

[2] D. H. Ballard, "Animate vision," *Artif. Intell.*, vol. 48, no. 1, pp. 57–86, 1991.

[3] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Conference*, 1988, pp. 147–152.

[4] D. Lowe, "Object recognition from local scale-invariant features," *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1150–1157 vol.2, 1999.

[5] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 5, pp. 530–535, May 1997.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] V. Ferrari, T. Tuytelaars, and L. Gool, "Simultaneous object recognition and segmentation from single or multiple model views," *Int. J. Comput. Vision*, vol. 67, no. 2, pp. 159–188, 2006.

[8] D. Lowe, "Local feature view clustering for 3d object recognition," *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–682–I–688 vol.1, 2001.

[9] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3d objects," *Int. J. Comput. Vision*, vol. 73, no. 3, pp. 263–284, 2007.

[10] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *Int. J. Comput. Vision*, vol. 66, no. 3, pp. 231–259, 2006.

[11] D. Kragic and M. Bjorkman, "Strategies for object manipulation using foveal and peripheral vision," in *ICVS '06: Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*. Washington, DC, USA: IEEE Computer Society, 2006, p. 50.

[12] S. Gould, J. Arfvidsson, A. Kaehler, B. Sapp, M. Messner, G. R. Bradski, P. Baumstarck, S. Chung, and A. Y. Ng, "Peripheral-foveal vision for real-time object recognition and tracking in video," in *IJCAI*, M. M. Veloso, Ed., 2007, pp. 2115–2121.

[13] 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.

[14] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of 5-th Berkeley Symposium on Mathmatical Statistics and Vision*.

[15] A. Noulas and B. Kröse, "Unsupervised visual object class recognition," *Advanced School of Computing and Imaging Conference, Lommel, Belgium*, 2006.

[16] D. Murray and J. J. Little, "Using real-time stereo vision for mobile robot navigation," *Auton. Robots*, vol. 8, no. 2, pp. 161–171, 2000.

[17] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[18] S. Heymann, K. Maller, A. Smolic, B. Froehlich, and T. Wiegand, "SIFT implementation and optimization for general-purpose GPU," *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2007.