Fast Adaptive Object Detection towards a Smart Environment by a Mobile Robot

Shigeru Takano, Ilya Loshchilov, David Meunier, Michèle Sebag, Einoshin Suzuki

Fast Adaptive Object Detection towards a Smart Environment by a Mobile Robot

Shigeru Takano[1], Ilya Loshchilov[2] David Meunier[2],
Michèle Sebag[2], and Einoshin Suzuki[1]

[1] Dept. Informatics, ISEE, Kyushu University
819-0395 Fukuoka, Japan
{takano,suzuki}@inf.kyushu-u.ac.jp
[2] TAO - CNRS & Univ. Paris-Sud
LRI, Bat. 490, Univ. Paris-Sud,
F-91405 Orsay, France
ilya.loshchilov@gmail.com, {david.meunier, michele.sebag}@lri.fr

Abstract. This paper proposes a novel method to detect objects by a mobile robot which adapts to an environment. Such a robot would help human designers of a smart environment to recognize objects in the environment with their attributes, which significantly facilitates his/her design. We first introduce Lifting Complex Wavelet Transform (LCWT) which plays an important role in this work. Since the LCWT has a set of controllable free parameters, we can design the LCWTs with various properties by tuning their parameters. In this paper we construct a set of LCWTs so that they can extract local features from an image by multi-scale. The extracted local features must be robust against several kinds of changes of the image such as shift, scale and rotation. Our method can design these LCWTs by selecting their parameters so that the mobile robot adapts to the environment. Applying the new set of LCWTs to the images captured by the mobile robot in the environment, a local feature database can be constructed. By using this database, we implement an object detection system based on LCWTs on the mobile robot. Effectiveness of our method is demonstrated by several test results using the mobile robot.

Keywords: adaptive object detection, keypoint detection, on-board robot vision, visual words, lifting complex wavelet transforms

1 Introduction

Recognizing objects in an environment with their attributes is an important first task for a designer of a smart environment. The task has been traditionally resolved by the designer manually, posing a considerable burden to him/her. Especially, it is necessary to estimate state attributes concerned with people and objects within the environment. These state attributes can be discriminative properties of the objects and the people, e.g., the object is still or moving, the person is walking or sitting. Typical conventional systems for ambient intelligence

have been developed with the sophisticated sensing technologies [5] with various sensors embedded in objects and clothing of people in the environment. We have witnessed a significant progress of autonomous mobile robots in these years [4], [7]. It is natural to use such a robot to fulfill the recognition task.

In this paper, we argue that the adaptation to the environment of the mobile robot is the key to success. Our mobile robot acquires a set of local features from images captured in the environments. A local feature is defined as an image pattern which differs from its immediate neighborhood, e.g. corners, blobs, edges [12]. Generally a lot of various local features can be extracted from an image including several objects. Our adaptation significantly reduces the computation time of the robot so that it operates in real time.

Our approach is to detect objects using Content Based Image Retrieval (CBIR) [10], which is to judge whether an object is known or unknown by whether a similar image exists in a database. So far visual features such as color, shape and textures of images, have been used in the CBIR systems. The most studied topic the CBIR at present is to extract local features such as SIFT [6] and SURF [1]. Although it has become possible to run SIFT and SURF algorithm on an ordinary PC, it is still required to develop faster method for extracting local features for onboard root vision. The paper [7] also proposed a method for object recognition by a mobile robot. However, since this method assumes smart objects equipped with wireless sensors, it is time consuming and cannot be used when a wireless sensor network is unavailable.

Our object detection is based on Lifting Complex Wavelet Transform (LCWT) [11], which is complex wavelet transform (CWT) [3] with controllable free parameters. We call them lifting parameters. The LCWTs within the lifting parameters can be adjusted so as to adapt to the environment. This means that our robots can acquire the local features efficiently since the LCWT can be trained so as to increase the useful local features for the object detection in number and to reduce the other features.

In the paper [11] we have introduced a set of LCWTs which can extract multi-scale features of the image. They are comparable to Difference of Gaussian (DoG) images which are used to extract SIFT features. By virtue of a fast wavelet transform algorithm, our LCWT can be computed faster than the DoG images computed by applying 2D Gaussian filters. Moreover, we can reduce the computation of local features in each scale. The reason is that because we only compute multi-scale components of the LCWT in the case in which the first components are greater than a specified threshold, which are obtained by applying the initial CWT to the image. The LCWT components in each scale can be easily computed by adding lifting terms, which include the lifting parameters, to the initial CWT components.

Using the multi-scale LCWT components we can detect keypoints which are robust for several kinds of changes of the image. Our method obtains a feature vector of 128 dimension in each keypoint of the image, which are computed by using the oriented gradients around it. In order to build our CBIR system, we extract all feature vectors from training images captured by the mobile robot,

and then cluster these vectors to form a vocabulary of visual words [9]. More recent works [8] successfully used k-means clustering but has difficulty in controlling the number of clusters. In this paper we use ϵ-mean clustering [2] to the set of feature vectors. Selecting ϵ parameter, we can tune the number of clusters which represent visual words. We expect that our CBIR system can retrieve similar images accurately as the entropy of the distribution of the feature vectors increases. Therefore we determine the lifting parameters so as to increase this entropy. Concretely we propose an algorithm for increasing the entropy based on genetic algorithm. The main contribution of our method is that the LCWTs can be designed so as to adapt to the environment for the mobile robot.

The outline of this paper is as follows. In Section 2, we give an overview of our adaptive object detection through CBIR using LCWT, and our robot architecture. Section 3 summarizes the LCWT and the multi-scale feature extraction. Section 4 presents algorithms for detecting and clustering of the keypoints in images, which are used in our CBIR system based on the LCWT, and shows how we design the adaptive LCWT to an environment. Section 5 contains experimental results. Concluding remarks and future work are given in Section 6.

2 Adaptive object detection for on-board robot vision

2.1 Adaptive object detection

As noted above, our goal is to detect objects by a mobile robot adapting to the environment. We design the LCWTs so as to adapt to images captured in the indoor environment. We first apply the LCWTs to the training images, and then obtain the feature vectors. We next build the visual vocabulary by applying the ϵ-means clustering to the set of feature vectors.

The visual vocabulary can be evaluated by the entropy of distribution of these vectors. We train the LCWTs by the evolutionary algorithm so as to increase this evaluation. By using the trained LCWT our robot can extract the existing or new object in the indoor environment. Figure 1 illustrates an overview of our adaptive object detection.

2.2 Robot architecture

Our robot moves in an indoor environment by two wheels and a supporting ball, and is equipped with a USB camera and an IR sensor. Since this camera and sensor can rotate from the left-hand side to the right-hand side and up and down by using three servos, our robot can capture images and measure the distance to obstacles in the area in front of the robot.

Our on-board robot vision is implemented on the Pandaboard, which is driven by the dual-core ARM Cortex-A9 OMAP4430, with each core running at 1 GHz, and equipped with 1 GB DDR2 RAM. A captured frame is in resolution of 320×240, and color depth of monochrome 8bit. The functions of robot vision can be operated on a Linux OS installed on the Pandaboard. Our mobile robot is shown in Fig.2.

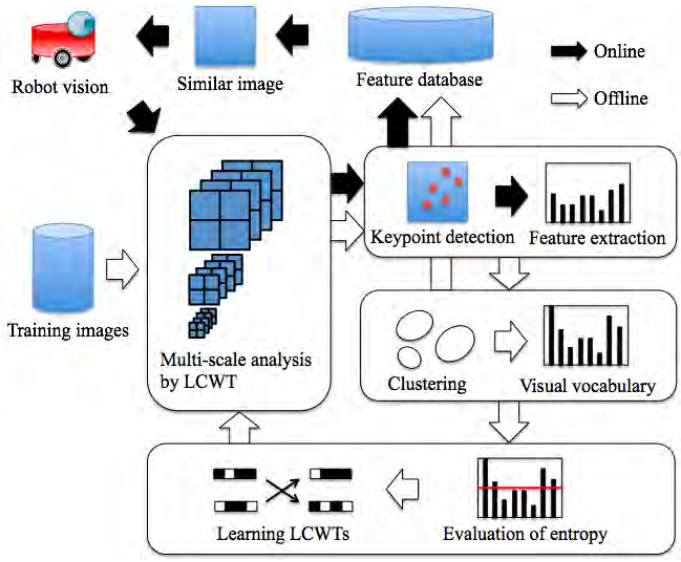


Fig. 1. Overview of adaptive object detection

3 Multi-scale feature extraction

In order to detect an object by robot vision, it is necessary to extract features fast from video frames captured by the mobile robot. This section introduces the LCWT [11] to extract multi-scale features of an image. The LCWT has controllable free parameters. We prepare a set of these lifting parameters for a multi-scale analysis of the image. Applying our LCWT to the video frames, we can obtain their multi-scale features fast.

3.1 Lifting Complex Wavelet Transform (LCWT)

Let us denote a set of complex wavelet filters by $\{h_k^{o,r}, g_k^{o,r}, h_k^{o,i}, g_k^{o,i}\}$ where h and g are lowpass and highpass filters, respectively. The superscript o means the initial filter, and r and i indicate that the corresponding filter are to obtain real and imaginary parts, respectively. Here we define a set of lifting complex wavelet filters $\{h_k^r, g_k^r, h_k^i, g_k^i\}$ as follows:

$$\begin{aligned} h_k^r &= h_k^{o,r}, \\ g_k^r &= g_k^{o,r} - \sum_m s_m h_{k-m}^{o,r}, \\ h_k^i &= h_k^{o,i}, \\ g_k^i &= g_k^{o,i} - \sum_m s_m h_{k-m}^{o,i}, \end{aligned} \tag{1}$$

where s represents the lifting parameters.

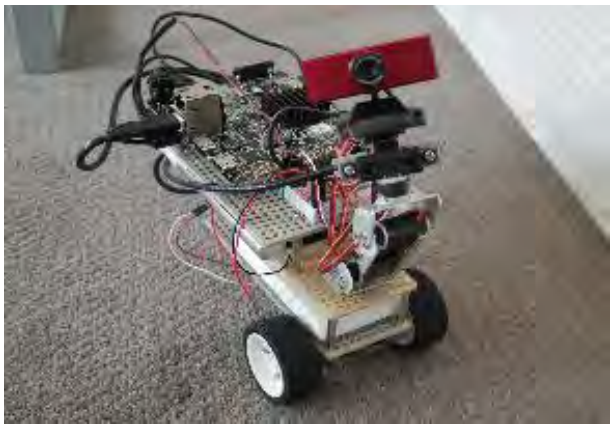


Fig. 2. Our robot.

Applying the lifting complex wavelet filters to image $C^1_{i,j}$ in row and column directions, we can obtain four components RR, RI, IR and II. In case of the RR, their LWTs are computed as follows:

$$C^0_{i,j} = \sum_{k,l} h^r_k h^r_l C^1_{2i+k,2j+l},$$
$$D^0_{i,j} = \sum_{k,l} h^r_k g^r_l C^1_{2i+k,2j+l},$$
$$E^0_{i,j} = \sum_{k,l} g^r_k h^r_l C^1_{2i+k,2j+l},$$
$$F^0_{i,j} = \sum_{k,l} g^r_k g^r_l C^1_{2i+k,2j+l}.$$

Since the filter g^r includes the lifting parameters, we can rewrite the above LWTs as

$$\begin{aligned}
C^0_{i,j} &= \hat{C}^0_{i,j},\\
D^0_{i,j} &= \hat{D}^0_{i,j} - \sum_m s_m \hat{C}^0_{i,j+m},\\
E^0_{i,j} &= \hat{E}^0_{i,j} - \sum_m s_m \hat{C}^0_{i+m,j},\\
F^0_{i,j} &= \hat{F}^0_{i,j} - \sum_m s_m \hat{D}^0_{i+m,j}\\
&\quad - \sum_m s_m \hat{E}^0_{i,j+m} + \sum_{m,m'} s_m s_{m'} \hat{C}^0_{i+m,j+m'},
\end{aligned}$$

where $\hat{C}^0$, $\hat{D}^0$, $\hat{E}^0$ and $\hat{F}^0$ are lowpass and three types of highpass components obtained by applying the initial filters to the input image C^1.

Similarly we can compute the LWTs in case of the RI, IR and II. The highpass components of six orientations can be computed by the combination of the RR, IR, IR and II [3].

$$\mathrm{Re}_{i,j} = \sqrt{RR_{i,j}^2 + II_{i,j}^2}, \tag{2}$$

$$\mathrm{Im}_{i,j} = \sqrt{RI_{i,j}^2 + IR_{i,j}^2}, \tag{3}$$

where $\mathrm{Re}_{i,j}$ and $\mathrm{Im}_{i,j}$ have each of 3 highpass components which differ in the feature of orientation. The paper [3] has shown that the robust features to the change of shift and rotation can be extracted by combining these components of the CWT. Finally we can obtain the robust features $H_{i,j}^0$ to the shift and rotation by summing these six highpass components of $\mathrm{Re}_{i,j}$ and $\mathrm{Im}_{i,j}$. In Fig. 3, we show an example of features obtained with LCWT.

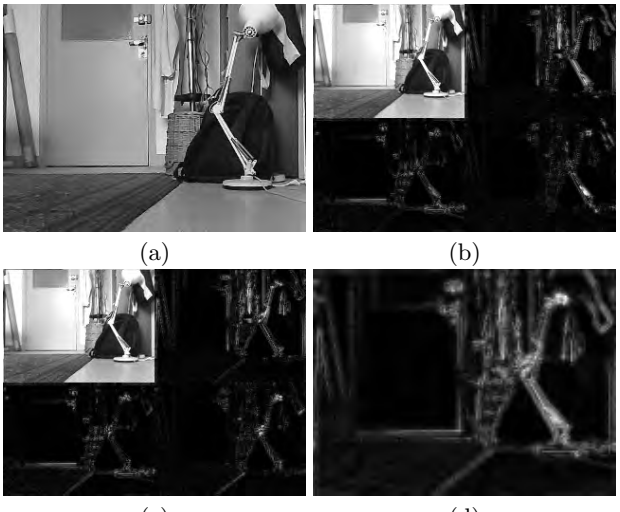


Fig. 3. An example of features obtained with LCWT: (a) input image, (b) LCWT of real parts, (c) LCWT of imaginary parts and (d) highpass components of LCWT

By iteratively applying the LCWT to the obtained each lowpass components C^0 in RR, RI, IR and II, a multi-resolution analysis can be conducted in the same way as the conventional wavelet transform. Therefore, we can obtain the lifting highpass components in each of resolutions $H_{i,j}^t, t = 0, -1, \cdots$ by using the LCWT, where t is an index that implies the resolution level.

3.2 Multi-scale analysis using LCWT

We must determine the parameters s_m in the LCWTs described in the previous subsection. In this paper we compute the set of the parameters so that the LCWT can extract the multi-scale local features of an image. We first determine the length of a lifting parameter. Concretely, we use the property that the length of the lifting highpass filters (1) depending on the length of the parameters s_m. As the initial filters we use the complex wavelet filters proposed in [3]. In order to extract multi-scale local features from images, we introduce the following four kinds of parameters as

$$
\begin{aligned}
&s_m^{(1)}, -1 \leq m \leq 1,\\
&s_m^{(2)}, -2 \leq m \leq 2,\\
&s_m^{(3)}, -4 \leq m \leq 4,\\
&s_m^{(4)}, -8 \leq m \leq 8.
\end{aligned}
$$

Using the parameters $s_m^{(\sigma)}$ $(\sigma = 1, \cdots, 4)$, we can construct four sets of the lifting complex wavelet filters $\{h_k^r, g_k^{\sigma,r}, h_k^i, g_k^{\sigma,i}\}$. The new highpass filters $g^{\sigma,r}$ and $g^{\sigma,i}$ can be written as

$$
\begin{aligned}
g_k^{\sigma,r} &= g_k^{o,r} - \sum_m s^{(\sigma)} h_{k-2m}^{o,r}, \quad \sigma = 1, \cdots, 4,\\
g_k^{\sigma,i} &= g_k^{o,i} - \sum_m s^{(\sigma)} h_{k-2m}^{o,i}, \quad \sigma = 1, \cdots, 4.
\end{aligned}
$$

In the next section, we will propose a method to determine the lifting parameters for adaptive object detection. In this section, each of the lifting parameters has the random value from -1 to 1 except for $m = 0$. Using $s_0^{(\sigma)}$ $(\sigma = 1, \cdots, 4)$, we tune the new highpass filters so as to satisfy the following conditions.

$$
\sum_k g_k^{\sigma,r} = 0, \quad \sum_k g_k^{\sigma,i} = 0, \quad \sigma = 1, \cdots, 4.
$$

We assume that these highpass filters can extract the multi-scale features from an image.

4 Adaptive object detection using LCWT

By tuning the lifting parameters, we can construct the LCWTs with various properties. In this paper, we concentrate to design the LCWT which can detect the local features from an image. We first introduce a keypoint detection algorithm using the LCWT. Each of the detected keypoints has a 128 dimensional vector each of which represents a feature based on the oriented gradients around it.

4.1 Keypoint detection and description by LCWT

As mentioned above, we can obtain the lifting highpass images $H_{i,j}^{t,\sigma}, t = 0, \cdots, -2$, $\sigma = 1, \cdots, 4$, where t and σ represent the multi-level parameter and the mutliscale parameter, respectively. We first extract the target point (i, j) as the candidate keypoints, if $H_{i,j}^{t,\sigma}$ is a local maximum or a minimum around the target point (i, j). The extracted candidate keypoints include a lot of unstable points. We therefore eliminate such unstable points which have low contrast or are poorly localized along an edge.

First, for each of the candidates, an interpolation of the nearby data [6] is used to accurately determine its position. The interpolation of nearby data is done by using the following Taylor expansion of the lifting highpass images

$$H(\mathbf{x}) = H + \frac{\partial H}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial H}{\partial \mathbf{x}}\mathbf{x}, \tag{4}$$

where H and its derivatives are evaluated at the candidate keypoint (i, j), and $\mathbf{x} = (y, x, \sigma)$ is the offset from this point. We can obtain the subpixel keypoint position by computing the extremum of (4). If the location of the extremum $\hat{\mathbf{x}}$ is larger than 0.5 in any dimension, the extremum lies closer to the other candidate keypoints. In this case, we reject this target keypoint. Otherwise the offset $\hat{\mathbf{x}}$ is added to its candidate keypoint to detect the subpixel keypoint position. Here if $H(\mathbf{x})$ at the subpixel position is lower than a specified threshold, then we also reject this point due to its low contrast.

Next step, we estimate whether the remaining points are corners. The lifting highpass images will have large values along edges even if the candidate keypoint is not robust for noise. Therefore, we eliminate the candidates keypoints on edges of which positions are imprecise. For imprecise peaks in the lifting highpass image, the principal curvature across the edge would be much larger than the principal curvature along it. These principal curvatures can be obtained as eigenvalues of the Hessian matrix of the lifting highpass image, by the same approach as SIFT [6]. The final keypoints are selected by performing the above localization process to the candidate keypoints. Figure 4 illustrates examples of the detected keypoints by the LCWT.

We extract local features at each of positions of the selected keypoints. In this subsection, we compute a descriptor vector for each keypoint. First, each keypoint is assigned orientations based on local image gradient directions. The magnitude and direction calculations for the gradient are done for every pixel in a neighboring region around the keypoint in the lowpass components.

$$m(u, v) = \sqrt{f_u(u, v)^2 + f_v(u, v)^2}, \tag{5}$$

$$\theta(u, v) = \tan^{-1}\left(\frac{f_u(u, v)}{f_v(u, v)}\right), \tag{6}$$

where

$$f_u(u, v) = C_{u+1,v}^t - C_{u-1,v}^t,$$
$$f_v(u, v) = C_{u,v+1}^t - C_{u,v-1}^t,$$

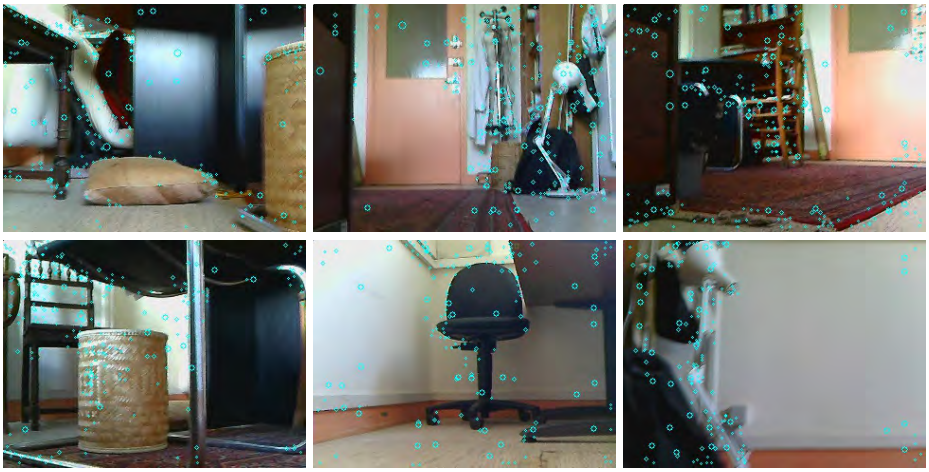


Fig. 4. Example of detected keypoints by LCWT

where $C^t_{u,v}$ is a mean of the lowpass component in (2) and (3) of resolution level t.

We make an orientation histogram with 36 bins, each bin covering 10 degrees. Each sample in the neighboring window added to a histogram bin is weighted by its gradient magnitude.

The window size depends on the length of the lifting highpass filters corresponding to the scale of the target keypoint. The peaks in this histogram can be regarded as principal orientations. We therefore assign the peaks of this histogram to the keypoint.

Here a descriptor vector is computed based on the magnitudes and orientations which are sampled around the keypoint location by (5) and (6). In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the orientation assigned at the keypoint. In each of 4×4 subregions of the region around the keypoint, the histogram with 8 orientation bins is created. Since there are $4 \times 4 = 16$ histograms each with 8 bins the vector has 128 elements. Finally the obtained vector is normalized to unit length in order to enhance invariance to affine changes in illumination.

4.2 Lifting parameters for object detection

In our previous work [11], the lifting parameters are computed by using training images to be the desired response in each of levels and scales. This paper determines the effective lifting parameters for the object detection. We first prepare a set of training images which are captured by our mobile robot in an office. Applying the LCWT to these training images, we can obtain a set of feature vectors in the keypoints of all images. Using ϵ-means clustering [2], these feature

vectors are classified into each cluster in which the Euclidean distance between any pair of feature vectors does not exceed a threshold ϵ. A pseudo code of the ϵ-means clustering is shown in Algorithm 1.

Algorithm 1 ϵ-means clustering

Input: ϵ, a threshold
Output: $\mathcal{C}$, the result of clustering
$\mathcal{C} = \emptyset$
for $t = 1$ **to** N_T
  $i(t) = \mathrm{argmin}_{c_j \in \mathcal{C}}\{d(x_t, c_j)\}$
  if $(d(x_t, c_i) > \epsilon)$ **then** $\mathcal{C} \leftarrow \mathcal{C} \cup (x_t, 1)$
  else $N_i + +$
  end if
end for

The result of the ϵ-means clustering is evaluated by the entropy of the distribution.

$$-\sum_k p(k) \log p(k).$$

Here $p(k)$ is a normalized distribution in the cluster k, which can be computed as $p(k) = N_k/N_T$, where N_k is the number of keypoints in the cluster k, and N_T is the number of all keypoints of the training images.

We wish to determine the lifting parameters so as to increase the entropy of the distribution. We propose an algorithm for determining the lifting parameters based on the Genetic Algorithm (GA). In this paper, a set of the lifting parameters $s^{(\sigma)}$ $(\sigma = 1, \cdots, 4)$ corresponds to an individual in GA, and is denoted with a binary array. We first choose the initial population of individuals. The initial lifting parameters are generated randomly by a method described in the previous subsection. We extract each distribution of feature vectors obtained by applying the LCWT with the set of parameters to the training images. Next, we evaluate the entropy of each distribution in that population. Here we repeat the following steps until a specified generation number N_g is reached.
1. Select the best-fit lifting parameters for reproduction.
2. Breed new lifting parameters through crossover and mutation operations (mutation probability $P_m = 0.01$).
3. Extract each set of feature vectors by applying new LCWTs to the training images.
4. Evaluate each entropy of the distribution obtained by using each of new LCWTs.
5. Replace least-fit population with new lifting parameters.
We show a pseudo code of the proposed algorithm in Algorithm 2.

Algorithm 2 Determining the lifting parameters based on the GA

```
1  Input: P, an initial set of lifting parameters, P_m, mutation probability
2  Output: the best lifting parameters in P
3  for i = 1 to N_g
4    e ← Evaluate(P)
5    P̃ ← ∅
6    for j = 1 to Size(P)
7      x ← Selection(P, e)
8      y ← Selection(P, e)
9      z ← Crossover(x, y)
10     if ( Random(0, 1) < P_m ) then
11       z ← Mutate(z)
12     end if
13     P̃ ← z
14   end for
15   P = P̃
16 end for
```

In step 4 of Algorithm 2, we evaluate the result of clustering by using the following estimation:

$$\mathbf{e} = \frac{-1}{\log K} \sum_{k=1}^{K} p(k) \log p(k),$$

where K is the number of clusters. We can avoid increasing the number of clusters by virtue of this estimation.

4.3 Adaptive object detection algorithm

Recent works in object based image retrieval have been proposed using the similarity of visual words [10]. We also make visual vocabulary for detecting the existing objects in an indoor environment. Each cluster $c_i \in \mathcal{C}$ represents a visual word, which is generated by ϵ-means clustering in the previous section.

An image is represented as a bag of visual words by clustering each keypoint feature into each visual word. In order to evaluate the similarity between an input image Q and each of database images T, we use the following histogram intersection [10].

$$S(Q, T) = \sum_{k=1}^{K} \min(h_q(k), h_t(k)),$$

where h_q and h_t are the distribution of the visual words corresponding to the input image and each of the database images, respectively. Our adaptive object detection judges whether existing or new objects by whether similar images exist in the database. Our robot carries out the adaptively object detection online by providing in advance that each set of LCWTs is trained in each of environment.

5 Simulations

5.1 Clustering of keypoint features

We first prepared the training images in order to learn the set of LCWTs adapting to the indoor environment. The training images were captured by the robot while our robot was moving around in the experimental room. A training image is of size 320×240 pixels, and in color depth of monochrome 8bit. Since the IR sensor of our robot can be rotated from the left-hand side to the right-hand side, the robot can measure the distances between the robot and the obstacles in the front, left and right directions. The robot moves so that their distances do not exceed a set of corresponding thresholds, $T_f = 300$mm, $T_l = 200$mm and $T_r = 200$mm. The robot therefore is expected to be able to avoid the obstacles in the room. Our robot controller is shown in Algorithm 3.

Algorithm 3 Robot controller

Input: T_f, T_l and T_r, thresholds for obstacle avoidance
do
  measure the distances d_{front}, d_{right} and d_{left}
  if ($d_{front} > T_f$) **then**
    if ($d_{left} > T_l$ & $d_{right} > T_r$) **then**
      capture a frame and then go forward
    else if ($d_{left} < T_l$ & $d_{right} > T_r$) **then** turn right
    else if ($d_{left} > T_l$ & $d_{right} < T_r$) **then** turn left
    else reverse
    end if
  else
    if ($d_{left} > T_l$ & $d_{right} > T_r$) **then**
      capture a frame and then reverse
    else if ($d_{left} < T_l$ & $d_{right} > T_r$) **then** turn right
    else if ($d_{left} > T_l$ & $d_{right} < T_r$) **then** turn left
    else reverse
    end if
  end if
loop until (robot receives a stop command)

Here we built the visual words by using 132 training images which include typical objects in the room. Figure 5 shows examples of the training images. We performed the ϵ-means clustering with $\epsilon = 2.0$ and trained the set of LCWT by using the proposed algorithm explained in the previous section. The entropies of the distribution of the feature vectors in the first generation and the 100th generation were 5.75 and 6.15, and their numbers of clusters were 1128 and 1434, respectively. In Fig. 6 (a) and (b), we illustrate the distributions in the first and 100th generations, and Fig. 6 (c) shows their cumulative density functions (cdf). Figure 6 (c) indicates that the distribution obtained by using the trained LCWTs is statistically close to the uniform distribution.

Fig. 5. Examples of the training images

We computed the visual words on the laptop PC with Intel Core i7 M640 2.80Ghz, 8.0GB of RAM. The computational time in each generation was about five minutes. This computational effort depends on the number of training images because all keypoints are updated in each generation.

5.2 Adaptive object detection by a mobile robot

Adaptive object detection by a mobile robot can be performed on images captured by our robot controller (Algorithm 3). By using the visual words generated in the previous subsection, we constructed the feature database from 3272 images captured in the experimental room.

Our detection algorithm for onboard robot vision can be done within one second by using the LCWTs trained in offline. Figure 7 shows examples of our object detection using the initial and the trained LCWTs. Our method works well for the CBIR effectively, since the target images are ranked higher by using the trained LCWTs than by the initial LCWTs, as shown in Fig. 7. These results indicate that our method can detect accurately the existing objects in the room. However the highly detailed image (e.g. rank 3 of Fig. 7 (a)) with many keypoints tends to be ranked higher than the expected target images. To avoid such a problem, we must consider the precision and recall for evaluation of CBIR, when the lifting parameters are computed.

6 Conclusion

We proposed a novel method for fast adaptive object detection for onboard robot vision by using the LCWT. Our method can design the LCWTs so as to adapt to the environment for a mobile robot. We clustered the feature vectors obtained by applying the LCWTs into the visual words, and built the visual vocabulary. In order to train the LCWTs we proposed the evolutionary algorithm so as to

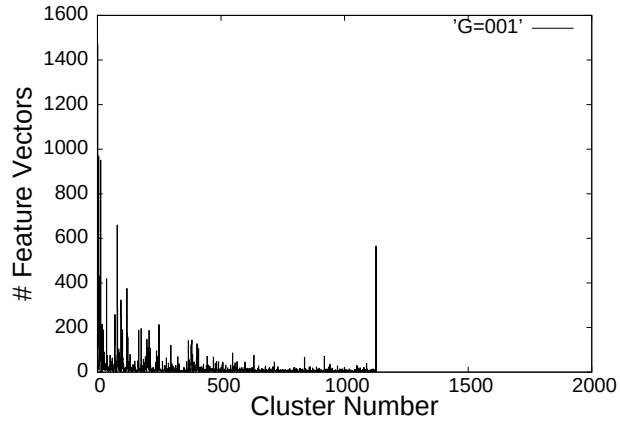
increase the entropy of distribution of these vectors. The experimental results indicate that the LCWT can be trained not only to extract local features from the image but also to retrieve the similar images from the database image accurately. In the future we will propose an online algorithm for learning the LCWT.

Acknowledgment

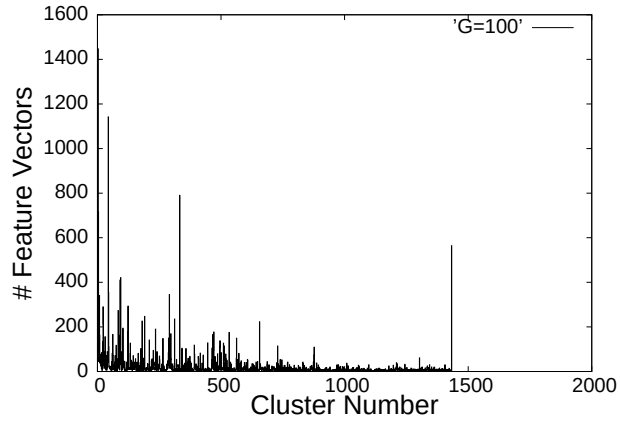
A part of this research was supported by Strategic International Cooperative Program funded by Japan Science and Technology Agency (JST) on the Japanese side and Agence Nationale de Recherches (ANR) on the French side. This work was partially supported by Grants-in-Aid for Scientific Research, JSPS (No. 23500183 and 24650070).

References

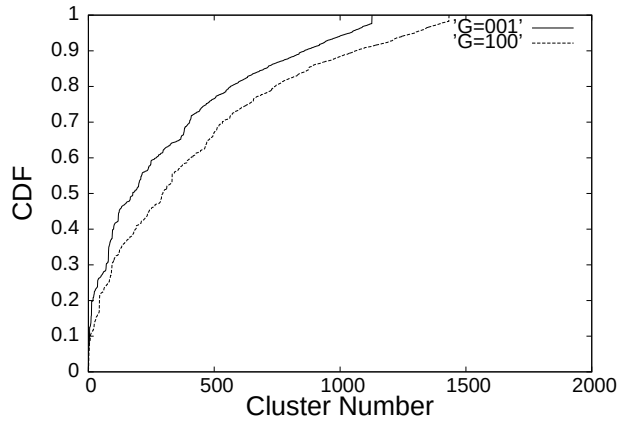
1. H. Bay, T. Tuytelaars and L. V. Gool, SURF: Speeded Up Robust Features, Proc. International Conference ECCV 2006, LNCS, vol. 3951, pp. 404-417, 2006.
2. P. Delarboulas, M. Schoenauer and M. Sebag, Open-Ended Evolutionary Robotics: An Information Theoretic Approach, Proc. PPSN (1), pp.334-343, 2010.
3. J. Fauqueur, N. Kingsbury and R. Anderson, Multiscale Keypoint Detection using the Dual-Tree Complex Wavelet Transform, Proc. IEEE Conference on Image Processing, pp. 8-11, 2006.
4. S. Frintrop, E. Rome and H. I. Christensen, Computational Visual Attention Systems and their Cognitive Foundation: A Survey, ACM Transactions on Applied Perception (TAP), vol. 7, no. 1, 2010.
5. J. IJsselmuiden, A.-K. Grosselfinger, D. Münch, M. Arens, R. Stiefelhagen, Automatic Behavior Understanding in Crisis Response Control Rooms, Proc. International Joint Conference on Ambient Intelligence (AMI 2012), LNCS, vol. 7683, pp. 97-112, 2012.
6. D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.
7. E. Menegatti, M.Danieletto, M. Mina, A.Pretto, A. Bardella, A. Zanella and P. Zanuttigh, Discovery, Localization and Recognition of Smart Objects by a Mobile Robot, Proc. Simulation, Modeling, and Programming for Autonomous Robots, LNCS, vol. 6472, pp. 436-448, 2010.
8. J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman, Object Retrieval with Large Vocabularies and Fast Spatial Matching Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007), pp. 1-8, 2007.
9. J. Sivic and A. Zisserman, Video Google: A Text Retrieval Approach to Object Matching in Videos, Proc. Ninth IEEE International Conference on Computer Vision (ICCV 2003), vol.2, pp. 1470-1477, 2003.
10. R. Szeliski, Computer Vision: Algorithms and Applications, Springer, New York, 2010.
11. S. Takano and E. Suzuki, New Object Detection for On-board Robot Vision by Lifting Complex Wavelet Transforms, Proc. Eleventh IEEE International Conference on Data Mining Workshops (ICDMW 2011), pp. 911-916, 2011.
12. T. Tuytelaars, K. Mikolajczyk, Local Invariant Feature Detectors: A Survey, Foundations and Trends in Computer Graphics and Vision, vol.3, no.3, pp. 177-280, 2007.



(a)



(b)



(c)

Fig. 6. Distribution of feature vectors

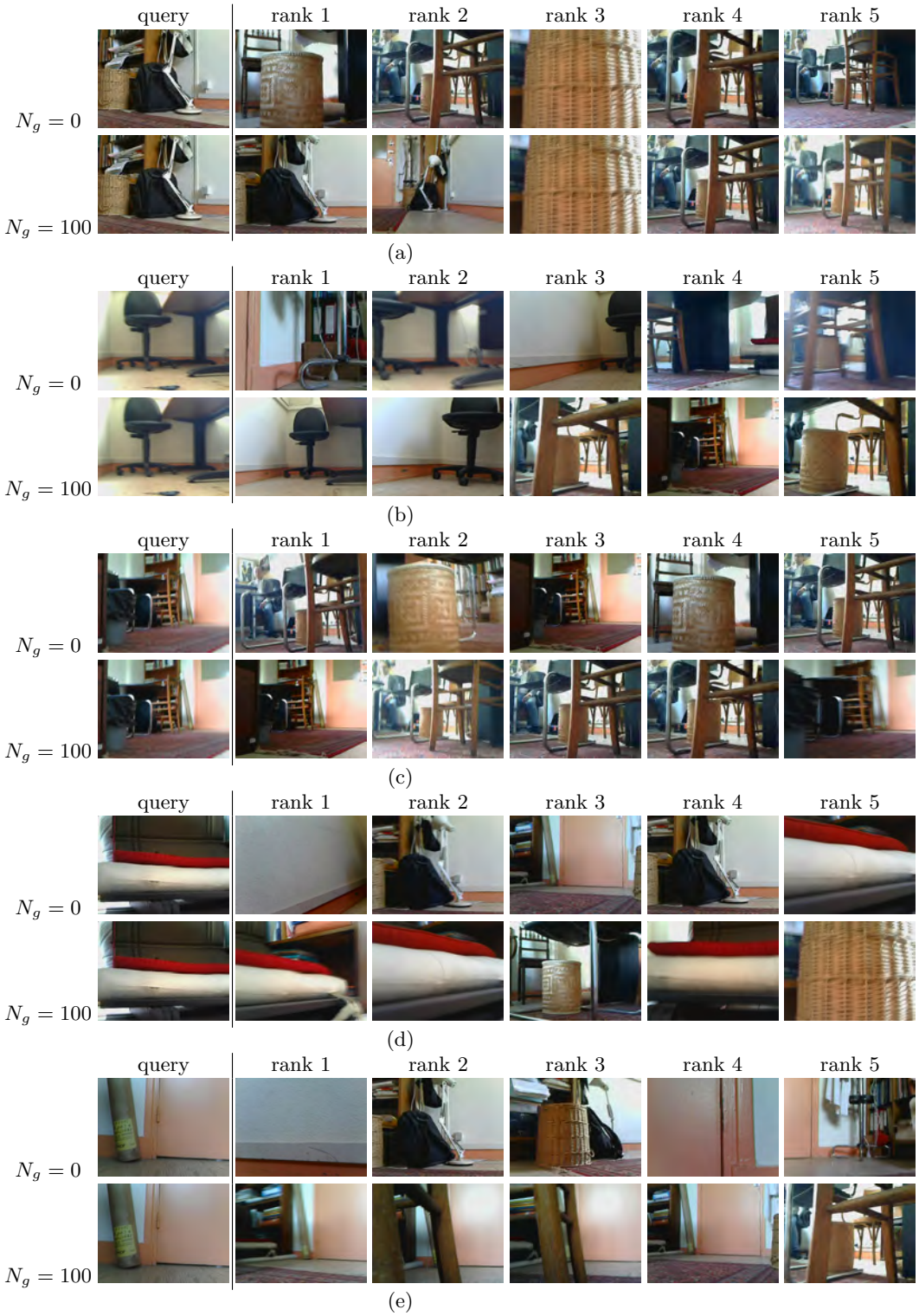


Fig. 7. Experimental results