2014 22nd International Conference on Pattern Recognition

3D-Object Recognition Based on LLC Using Depth Spatial Pyramid

Toru Nakashika, Takafumi Hori
Graduate School of System Informatics
Kobe University
1-1 Rokkodai, Kobe, Japan
nakashika@me.cs.scitec.kobe-u.ac.jp, hori@me.cs.scitec.kobe-u.ac.jp

Tetsuya Takiguchi, Yasuo Ariki
Organization of Advanced Science and Technology
Kobe University
1-1 Rokkodai, Kobe, Japan
takigu@kobe-u.ac.jp, ariki@kobe-u.ac.jp

***Abstract*—Recently introduced high-accuracy RGB-D cameras are capable of providing high quality three-dimension information (color and depth information) easily. The overall shape of the object can be understood by acquiring depth information. However, conventional methods adopted this camera use depth information only to extract the local feature. To improve the object recognition accuracy, in our approach, the overall object shape is expressed by the depth spatial pyramid based on depth information. In more detail, multiple features within each sub-region of the depth spatial pyramid are pooled. As a result, the feature representation including the depth topological information is constructed. We use histogram of oriented normal vectors (HONV) designed to capture local geometric characteristics as 3D local features and locality-constrained linear coding (LLC) to project each descriptor into its local-coordinate system. As a result of image recognition, the proposed method has improved the recognition rate compared with conventional methods.**

I. Introduction

Object recognition means that the computer recognizes objects from real world images by their names. It is one of the most challenging tasks in the field of computer vision. There are two types of object recognition tasks: instance recognition and category recognition. Instance recognition is to recognize known object instances. On the other hand, category recognition is to determine the category name of an unknown object. Regarding the achieving of human-like vision by a computer, it is expected that any such technology will be applied to robotic vision. Recently-introduced high-accuracy RGB-D cameras into its field are capable of providing high quality three dimension information (color and depth). Thus, this paper proposes a method for multi-class object image classification using 3D information (see Fig. 1).

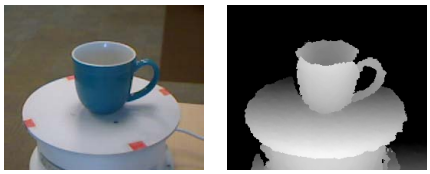

Fig. 1. RGB-D information

Recent image classification systems mainly consist of the following three parts: feature extraction using scale-invariant feature transform (SIFT) [1], coding scheme using bag-of-features (BoF) [2] and pooling process using spatial pyramid matching (SPM) [3]. The BoF method for characterizing the entire image uses the appearance frequency histogram of the localizing features. This feature is especially robust against spatial translations of features, although the robustness leads to disregard of location information.

SPM is used as extensions of the BoF. The method partitions the image into hierarchical spatial sub-regions and computes histograms of local features from each sub-region, as shown in Fig. 2 (typically, $2^l \times 2^l$ subregions, $l = 0, 1, 2$). This spatial pyramid restricted by position has shown very promising performance on many image classification tasks. These techniques used for 2D images is applied to 3D object recognition without any changes. For that reason, even though the depth information captures the overall shape of an object, conventional methods use depth information only to extract the local feature.

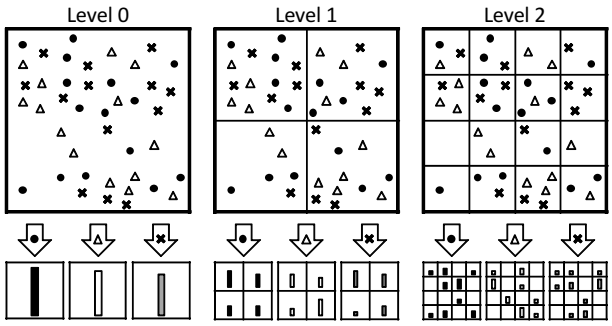

Fig. 2. Spatial Pyramid Matching

To deal with this problem, in our proposed approach, the overall object shape is expressed by the depth spatial pyramid based on depth information. In more detail, multiple features within each sub-region of the depth spatial pyramid are pooled. As a result, the feature representation including the depth topological information is constructed. We use not only SIFT, but also histograms of oriented normal vectors (HONV) designed to capture local geometric characteristics. We also adopt locality-constrained linear coding (LLC), which utilizes local constraints to project each descriptor into its local-coordinate system.

This paper is organized as follows: in Sections 2, 3, 4, and 5, the proposed method is described. In Section 6, the

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.724

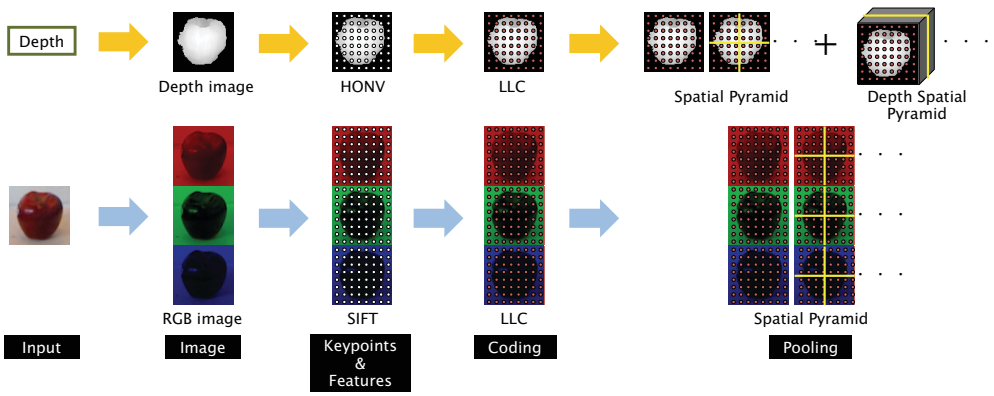

Fig. 3. System overview

performance of the proposed method is evaluated. Section 7 provides a summary and discusses future work.

II. Overview of the Proposed Method

Fig. 3 shows the system overview. First, the depth image and the RGB images of each channel is created from depth and color information. Feature points of each image are located by grid sampling, and features (HONV and SIFT) are extracted from each feature point. HONV is extracted from the depth image and SIFT is extracted from the RGB images. The extracted features are coded using LLC. Then, multiple codes within each sub-region of the spatial pyramid are pooled together. The pooling of the depth spatial pyramid is additionally processed for the depth image. Finally, the pooled features from all sub-regions are concatenated together for classification. The classifier is trained by this concatenated vector of training images. The test data is classified by the trained classifier and the recognition result is output. In the following sections, each process in the proposed method is described in detail.

III. Histogram of Oriented Normal Vectors (HONV)

The HONV is local features using depth information, which is designed to capture local geometric characteristics for object recognition [4]. The local 3D shape characteristics are represented as a local distribution of a normal vector orientation at every surface point.

The depth information captured by a depth sensor is converted to the depth image. We denote each pixel in the depth image as $p = (x, y, d(x, y))$. $d(x, y)$ shows the distance between the pixel position and the sensor, that is, depth information. The normal vector at pixel p is computed by

$$\boldsymbol{N} = (-\frac{\partial d(x,y)}{\partial x}, -\frac{\partial d(x,y)}{\partial y}, 1) \tag{1}$$

where $\frac{\partial d(x,y)}{\partial x}$ and $\frac{\partial d(x,y)}{\partial y}$ are calculated using the finite difference approximation, respectively. The zenith angle θ and the azimuth angle φ of spherical coordinates are used as normal vector orientation. Fig. 4 shows the relationships among a normal vector, the zenith angle, and the azimuth angle. Each angle is computed as

$$\theta = \tan^{-1}\left(\left(\frac{\partial d(x,y)}{\partial x}\right)^2 + \left(\frac{\partial d(x,y)}{\partial y}\right)^2\right)^{\frac{1}{2}} \tag{2}$$

$$\varphi = \tan^{-1}\left(\frac{\partial d(x,y)}{\partial y} \Big/ \frac{\partial d(x,y)}{\partial x}\right) \tag{3}$$

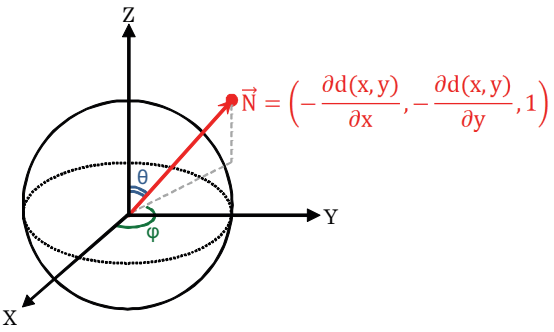

Fig. 4. Zenith angle θ and azimuthal angle φ of a normal vector

These two angles are used to capture the overall shape of an object in three-dimensions in a similar way to histograms of oriented gradients (HOG) [5] in two-dimensions. Firstly, the image is divided into cells. The center of each cell is the feature point. For each cell, the normal vector orientation at each pixel is quantized and voted into a 2D histogram of θ and φ. To restrain spatial boundary effects, a 2D Gaussian smoothing process is employed over adjacent cell histograms. The 2D histogram is used as HONV feature of the corresponding feature point.

IV. Locality-constrained Linear Coding (LLC)

In this paper, we adopted LLC as the coding scheme [6]. $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$ denotes a set of D-dimensional local features x_i extracted from an image. Similarly, a codebook $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_M] \in \mathbb{R}^{D \times M}$ denotes a set of M codewords b_j generated by using K-Means algorithm. Coding schemes finally convert each feature into

a M-dimensional code. We show the LLC method comparing with two existing coding schemes in Fig. 5.

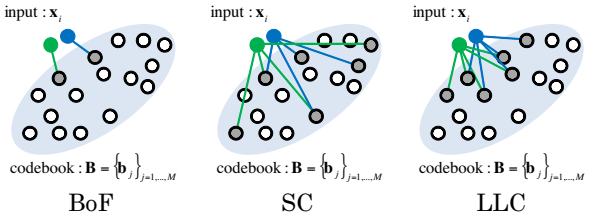

Fig. 5. Comparison between BoF, SC and LLC

A. Coding descriptors in BoF

The BoF method is synonymous with solving the following constrained least square fitting problem:

$$\arg\min_{C} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{c}_i\|^2 \tag{4}$$
$$s.t. \|\boldsymbol{c}_i\|_{l^0} = 1, \|\boldsymbol{c}_i\|_{l^1} = 1, \boldsymbol{c}_i \geq 0, \forall i$$

where $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N]$ is a set of codes for $\boldsymbol{X}$. The cardinality constraint $\|\boldsymbol{c}_i\|_{l^0} = 1$ means that each code $\boldsymbol{c}_i$ contains only one non-zero element, corresponding to the quantization id of $\boldsymbol{x}_i$. $\|\boldsymbol{c}_i\|_{l^1} = 1, \boldsymbol{c}_i \geq 0$ means that the coding weight for $\boldsymbol{x}$ is 1. This process can be reagarded as searching the nearest neighbor.

B. Coding descriptors in ScSPM

In BoF, because each feature is represented by a single codebook, the large quantization errors can occur, as shown in Fig. 5. To improve this loss, a sparsity regularization term of l^1 norm is used instead of the restrictive cardinality constraint $\|\boldsymbol{c}_i\|_{l^0} = 1$ in Eq. (4) [7]. As a result of this modification, a feature is represented by plural codebooks. Thus coding each local feature $\boldsymbol{x}_i$ becomes a standard sparse coding (SC) problem [8].

$$\arg\min_{C} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{c}_i\|^2 + \lambda\|\boldsymbol{c}_i\|_{l^1} \tag{5}$$

The quantization error can be greatly decreased by introducing this sparsity regularization term.

C. Coding descriptors in LLC

Generally, locality is more useful than sparsity. LLC introduces a locality constraint instead of the sparsity constraint in Eq. (5). In a word, the input feature is expressed by the codebooks of its neighborhood. The LLC code is computed as follows:

$$\arg\min_{C} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{c}_i\|^2 + \lambda\|\boldsymbol{d}_i \odot \boldsymbol{c}_i\|^2 \tag{6}$$
$$s.t. \mathbf{1}^\top \boldsymbol{c}_i = 1, \forall i$$

where $\odot$ denotes the element-wise multiplication, and $\boldsymbol{d}_i \in \mathbb{R}^M$ is the locality adapter defined as

$$\boldsymbol{d}_i = \exp\left(\frac{D(\boldsymbol{x}_i, \boldsymbol{B})}{\sigma}\right) \tag{7}$$

where $D(\boldsymbol{x}_i, \boldsymbol{B}) = [D(\boldsymbol{x}_i, \boldsymbol{b}_1), \ldots, D(\boldsymbol{x}_i, \boldsymbol{b}_M)]^T$, and $D(\boldsymbol{x}_i, \boldsymbol{b}_j)$ is the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{b}_j$. σ is used for adjusting the weight decay speed for the locality adapter.

To improve the recognition performance, the coding scheme needs to generate similar codes for similar features. However, as shown in Fig. 5, the SC process might make the similar feature expressed in different codebooks. Thus, there is the possibility that the correlation between the codes is lost. In contrast, the locality adapter in LLC converts similar features into similar codes. The LLC code captures the correlations between similar features by sharing codebooks.

V. Spatial Pooling

Spatial pooling is the process of dividing an image into sub-regions, and pooling multiple features within each sub-region. In this paper, we use spatial pyramid ($2^l \times 2^l$ subregions) for each image in 2D. Depth spatial pyramid is additionally used to divide each depth image in 3D.

A. Depth spatial pyramid

The depth spatial pyramid is a spatial pyramid in the depth coordinate calculated from depth information. Assuming that the depth value is a coordinate, we divide the depth image into sub-regions. However, the measured depth values are unreliable and disperse unlike coordinates of a common spatial pyramid. If the space is simply divided equally like the spatial pyramid, the numbers of feature points within each sub-region are biased. Therefore, we divide it into sub-regions including equal number of points without dividing by coordinates. Typically, m subregions ($m = 0, 1, 2$) are used (Fig. 6). 3D space of an object is spatially constrained by using the depth spatial pyramid and the spatial pyramid together. As a result, the overall 3D shape of the object can be expressed.

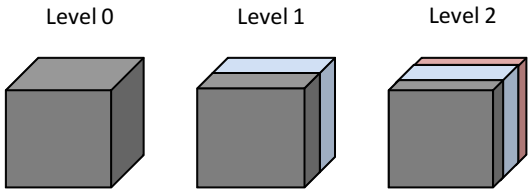

Fig. 6. Depth Spatial Pyramid

B. Pooling method

In each spatial pyramid, multiple codes within each sub-region are pooled together. These pooled features from each sub-region are concatenated and normalized as the final image feature representation. We use max pooling as pooling method:

$$\boldsymbol{c}_{out1} = \max(\boldsymbol{c}_{in1}, \ldots, \boldsymbol{c}_{inH}) \tag{8}$$

where H denotes a number of feature points within the sub-region, and the max function in a row-wise manner returns a vector with the same size as $\boldsymbol{c}_{in1}$. These pooled features $\boldsymbol{c}_{out1}$ are concatenated as the feature vector $\boldsymbol{c}_{in}$. It is normalized by

$$\boldsymbol{c}_{out2} = \boldsymbol{c}_{in}/\|\boldsymbol{c}_{in}\|_2. \tag{9}$$

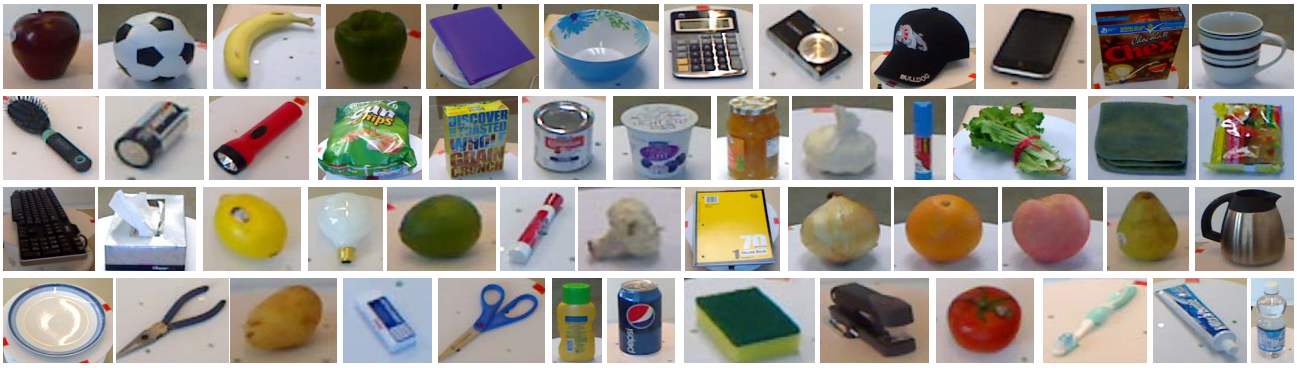

Fig. 7. Objects from the RGB-D Object Dataset

TABLE I. RECOGNITION RESULTS AND COMPARISONS(%)

	Category			Instance		
	RGB	**Depth**	**RGB-D**	**RGB**	**Depth**	**RGB-D**
ICRA11[9]	74.3 ± 3.3	53.1 ± 1.7	81.9 ± 2.8	59.3	32.3	73.9
Kernel desc[10]	80.7 ± 2.1	80.3 ± 2.9	86.5 ± 2.1	90.8	**54.7**	91.2
CKM desc[11]	N/A	N/A	86.4 ± 2.3	82.9	N/A	90.4
HMP[12]	74.7 ± 2.5	70.3 ± 2.2	82.1 ± 3.3	75.8	39.8	78.9
ISER12[13]	82.4 ± 3.1	81.2 ± 2.3	87.5 ± 2.9	92.1	51.7	92.8
Proposed	**85.3 ± 1.6**	**82.9 ± 2.3**	**89.2 ± 1.6**	**93.4**	42.5	**94.2**

This c_{out2} is the final image feature representation. Here, for the depth image, we concatenate two feature representations made from spatial pyramid and depth spatial pyramid.

VI. EXPERIMENTAL EVALUATION

In this section, we evaluate our proposed method, comparing with conventional methods using a benchmark dataset.

A. Experimental Conditions

We used the RGB-D Object Dataset for the experiments [14]. It is composed of 300 objects, 51 categories and about 42,000 images containing RGB and depth information. Each object is recorded from three viewing heights (30°, 45° and 60° angles) while it rotates on a turntable. Fig. 7 shows the examples of each category. For our experiments, we used the same setup as in [9], distinguishing between category and instance recognition. Firstly, category-level classification experiments were conducted with 51 class labels. We randomly selected one object from each category for the test, and trained the classifier on the remaining objects. The averaged accuracy and the standard deviation over 10 random trials are reported for category recognition. Secondly, instance classification experiments with 300 objects were conducted. We trained the classifier on the images captured from 30° and 60° elevation angles, and tested them on the images of the 45° angle. We present object recognition results on the RGB-D Object dataset with only depth features (Depth), only color features (RGB), and with both depth and color features (RGB-D). In our setup, the SIFT and the HONV features were extracted from points densely located by every 4 pixels on an image, under three scales, 8 × 8, 12 × 12 and 16 × 16, respectively. The codebook size was 1024. We used as the classifier multi-class SVM (linear) to classify the vectors.

B. Experimental Results and Discussion

Table I shows the recognition results and the comparison with conventional methods. From Table I, it can be confirmed that the proposed method improved the accuracy. This result shows the effectiveness of the proposed method using HONV, LLC and depth spatial pyramid. However, only the result with depth features (Depth) for instance recognition does not outperform other methods. This is because the dataset contains many objects having the same shapes but the colors are different. It is generally difficult to recognize those objects only with shape information. Especially, the proposed method specialized for the shape representation was strongly influenced, and therefore its recognition accuracy was not improved. Nevertheless, the depth information contributes to the improvement of the recognition rate when mixed up with the color information (RGB-D).

VII. CONCLUSION

This paper presented a 3D object recognition method using HONV, LLC and depth spatial pyramid based on depth information. The feature representation including the topological information of shape was constructed by using depth spatial pyramid and spatial pyramid together. Our proposed method of expressing overall object shapes demonstrated the better performance compared with conventional methods in the experiments using 3D object dataset. In the future, we will study the sub-region division method of more effective spatial pyramid and the pooling method.

REFERENCES

[1] D. G. Low, "Distinctive image features from scale-invariant keypoints," Journal of Computer Vision, Vol. 60, No. 2, pp. 91–110, 2004.

[2] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, "Object Recognition as Machine Translation: Learning aLexicons for a Fixed Image Vocabulary," ECCV Workshop on Statistical Learning in Computer Vision, pp. 1–22, 2004.

[3] S. Lazebnik, C. Schmid and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 2169–2178, 2006.

[4] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of Oriented Normal Vectors for Object Recognition with a Depth Sensor," The Asian Conference on Computer Vision, 2012.

[5] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893, 2005.

[6] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang and Y. Guo, "Locality-constrained Linear Coding for Image Classification," IEEE Conference on Computer Vision and Pattern Recognition, pp. 3360–3367, 2010.

[7] J. Yang, K. Yu, Y. Gong and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," IEEE Conference on Computer Vision and Pattern Recognition, pp. 1794–1801, 2009.

[8] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," Advances in Neural Information Processing Systems, MIT Press, pp. 801–808, 2006.

[9] K. Lai, L. Bo, X. Ren, and D. Fox, "A Large-Scale Hierarchical Multi-View RGB-D Object Dataset," IEEE International Conference on Robotics and Automation, pp. 1817–1824, 2011.

[10] L. Bo, X. Ren and D. Fox, "Depth Kernel Descriptors for Object Recognition," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 821–826, 2011.

[11] M. Blum, J. Springenberg, J. Wlfing, and M. Riedmiller, "A Learned Feature Descriptor for Object Recognition in RGB-D Data," IEEE International Conference on Robotics and Automation, pp. 1298–1303, 2012.

[12] L. Bo, X. Ren, and D. Fox, "Hierarchical Matching Pursuit for Image Classification: Architecture and Fast Algorithms," Neural Information Processing Systems (NIPS), 2011.

[13] L. Bo, X. Ren, and D. Fox, "Unsupervised Feature Learning for RGB-D Based Object Recognition," In International Symposium on Experimental Robotics, 2012.

[14] RGB-D Object Dataset, http://www.cs.washington.edu/rgbd-dataset/