Journal of Southeast University (English Edition) Vol. 28, No. 4, pp. 404 - 409 Dec. 2012 ISSN 1003—7985

Online object detection and recognition using motion information and local feature co-occurrence

Zhang Suofei[1] Filliat David[2] Wu Zhenyang[1]

([1]Key Laboratory of Underwater Acoustic Signal Processing of Ministry of Education, Southeast University, Nanjing 210096, China)
([2]UEI, ENSTA ParisTech, Paris 91762, France)

Abstract: An object learning and recognition system is implemented for humanoid robots to discover and memorize objects only by simple interactions with non-expert users. When the object is presented, the system makes use of the motion information over consecutive frames to extract object features and implements machine learning based on the bag of visual words approach. Instead of using a local feature descriptor only, the proposed system uses the co-occurring local features in order to increase feature discriminative power for both object model learning and inference stages. For different objects with different textures, a hybrid sampling strategy is considered. This hybrid approach minimizes the consumption of computation resources and helps achieving good performances demonstrated on a set of a dozen different daily objects.

Key words: object recognition; online learning; motion information; computer vision

doi: 10.3969/j. issn. 1003 - 7985. 2012. 04. 006

Humanoid robots are drawing an increasing amount of interest both in scientific and commercial communities. These robots can interact with non-expert human in a context of domestic services or entertainments. In this paper, we are specifically interested in small entertainment robots of humanoid or animal shape, and only visual cues are taken into account. When the user wants the robot to memorize something in the environment, a necessary operation is just waving around the object in front of the robot and notifying the name of the target to the robot, e. g., by vocal signal. Our algorithm uses the motion information over consecutive frames to implement a coarse image segmentation which extracts the approximate position of the object from a cluttered background. Then the corresponding model of this object is built progressively over frames by our tracking mechanism. The integration of object tracking and recognition helps the robot learn to recognize objects with less intervention from humans.

In the inference stage of our system, we concentrate on the detection and recognition of the object on a single frame rather than on several consecutive frames. As a consequence, the inference algorithm can be solely deployed on some key frames to adapt the applications with realistic computation constraints and achieve a real-time online system. Meanwhile, it also means that the algorithm can detect the object from a background even it is static.

The frame of our work is based on the "bag of visual words" method[1]. The features are clustered into a codebook and used without taking their position into account. The four main implementation choices in such a frame are how to sample features, how to describe them, how to characterize the resulting distributions and how to classify images based on the results.

About the former two parts, numerous works in object recognition and image understanding are based on the scale invariant feature transform (SIFT) method because of its remarkable quality and simplicity to use. Since most local feature extraction methods work on the gray scale image[2-3], color information is often injected into the descriptor[4] for dealing with the color image.

In our work, from the perspective of speed, we choose the speeded up robust features (SURF) method[3] as the feature detector and descriptor. Since we want to tackle the information in video sequences efficiently, the SURF method provides us a high speed solution with acceptable quality. We also meet the same typical sparse features problem on some consistently colored objects as other relevant works[5]. To solve this problem, we adopt the superpixel dense sampling strategy[4] as the complement to the standard SURF process.

About the image classification stage, Refs. [6 - 7] proposed a dictionary construction method which is similar to Ref. [8] but simpler and fully incremental. However, the static dictionary construction method used in this paper is based on Ref. [9].

1 Object Detection

1.1 Bag of visual words method

Bag of visual words is a popular method for image categorization. The visual words used in our work are based on SURF descriptors. We use the vocabulary tree ap-

Received 2012-06-05.

Biographies: Zhang Suofei (1982—), male, graduate; Wu Zhenyang (corresponding author), male, doctor, professor, zhenyang@ seu. edu. cn.

Foundation item: The National Natural Science Foundation of China (No. 60672094, 60971098).

Citation: Zhang Suofei, Filliat David, Wu Zhenyang. Online object detection and recognition using motion information and local feature co-occurrence[J]. Journal of Southeast University (English Edition), 2012, 28 (4): 404 - 409. [doi: 10.3969/j. issn. 1003 - 7985. 2012. 04. 006]

proach [9] where a tree is constructed by applying k-means at each level, thus hierarchically segmenting the feature space. The output of the discretization is the dictionary. Given an image, a vector of features $\boldsymbol{F} = \{f_1, f_2, \ldots, f_{N_F}\}$ is extracted to represent the image, where N_F is the amount of features in the current image. The features are assigned into the dictionary to obtain the vector of visual words $\boldsymbol{V} = \{v_1, v_2, \ldots, v_{N_F}\}$, where v_i is the index of the visual word in dictionary. Rewriting $\boldsymbol{V}$ as the histogram of visual words $\boldsymbol{V}_h = \{c_1, c_2, \ldots, v_N\}$, where N is the size of vocabulary and c_i is the number of occurrences of visual word v_i. By this process the spatial structure of the image is omitted and the statistic information is extracted.

1.2 Feature extraction

The SURF method uses a fast-Hessian detector to find keypoints and creates a descriptor of the region around every keypoint. One typical problem associated with such sparse feature-based method is the scarcity of descriptions of textureless regions. To solve this problem, we use a dense sampling strategy named superpixel.

Superpixel is an over-segmentation of images that is assumed to be consistent with object boundaries but breaks large objects into small pieces. The method in our work is inspired by Ref. [4]. The boundaries of superpixels are obtained by watersheds on a negative absolute Laplacian image with LoG extremas as seeds. After segmenting, we compute a SURF descriptor at the center of each superpixel. Fig. 1(e) illustrates the points where the SURF descriptors are computed by our superpixel method.

The descriptor of the SURF method is a 64-dimensional vector without color information. In our work we add color information in the case of superpixels by calculating the mean color over superpixel pixels and add the channel h of HSV color space to the vector for describing the general color information of this region. So the feature descriptor $\boldsymbol{f}$ we used is a 65-dimensional vector $\boldsymbol{f} = \{x_1, x_2, x_3, \ldots, x_{65}\}$, $x_i \in [0, 256]$. The definition of distance between two descriptor vectors is

$$d_{ss}(f^a, f^b) = \frac{\sqrt{\left(p \sum_{i=1}^{64} (x_i^a - x_i^b)^2\right) \Big/ 64 + (1-p)(x_{65}^a - x_{65}^b)^2}}{256} \tag{1}$$

Here the proportion p controls the influence of color information on the whole method. The reference of color information as well as the conventional SURF descriptor helps us discriminate targets in different colors.

Applying this method on well textured objects is useless as these objects are well recognized from SURF keypoints. In order to reduce computational requirements, we therefore use a saliency measure from Ref. [10] as the criterion to automatically decide if the SURF keypoints are sufficient to correctly characterize the current object. The saliency of the target model is calculated as

$$S_t = \frac{1}{S} \sum_{i=1}^{S} \sum_{j=1}^{S} P(v_j \mid v_i) \tag{2}$$

where S is the number of SURF features in the model; $P(v_j \mid v_i)$ is the conditional probability of observing visual word v_j given the visual word v_i. The summation can be computed directly from matrix $\boldsymbol{C}$ which is obtained in Section 1.3. Our definition is a variation by the hard assignment rule to the original definition in Ref. [10]. This saliency takes the contextual information around every feature into account and describes the complexity of the current model quantitatively. When the system notices the model lack of features, it extends the training stage and switches to the superpixel-SURF in order to obtain a dense sampling of the target.

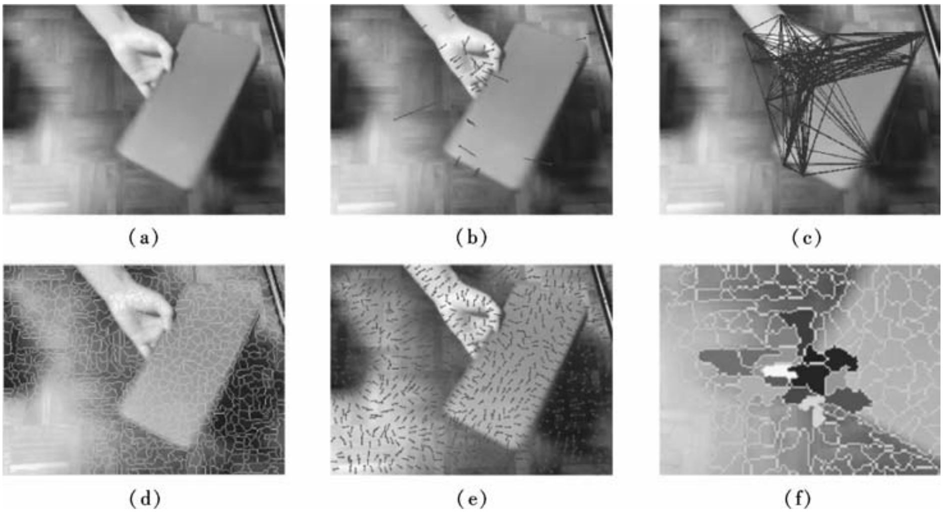

Fig. 1 Illustration of SURF and superpixel-SURF methods. (a) Original image; (b) SURF features; (c) SURF defined pairwise features; (d) Superpixels; (e) Superpixel-SURF features; (f) Superpixel defined co-occurrence

1.3 Co-occurrence between features

Instead of using visual words vectors as the representation of an image directly, we choose co-occurrence between visual words as the element in the bag of words frame. There are two definitions of co-occurrence for SURF and superpixel-SURF, respectively. For every extracted feature $\boldsymbol{f}_i$ in the SURF frame, we search n most neighboring features $\boldsymbol{f}_j \in \{f_1, f_2, \ldots, f_n\}$ and define each pair $\{\boldsymbol{f}_i, \boldsymbol{f}_j\}$ as the co-occurrence of two features (see Fig. 1(c)). For superpixel-SURF features, we implement a 2-depth neighborhood of superpixels as the definition of the co-occurrence which is similar to Ref. [4]. As illustrated in Fig. 1(f), the superpixels sharing the boundary with superpixel i (the white block in the center of the area) are called direct neighbors of superpixel i. Moreover, neighbors of these direct neighbors are called 2-depth neighbors to superpixel i. All the features extracted from the 2-depth neighbors $\boldsymbol{f}_j \in \{f_1, f_2, \ldots, f_n\}$ are paired to the feature $\boldsymbol{f}_i$ as the co-occurrence $\{\boldsymbol{f}_i, \boldsymbol{f}_j\}$. When the user is moving the object, a visual word co-occurrence matrix $\boldsymbol{C}$ with a dimension $N \times N \times N_{\mathrm{L}}$ is learned over frames, where v_i is the visual word of $\boldsymbol{f}_i$; $\boldsymbol{C}[v_i, v_j]$ is the normalized count of the co-occurrence of visual words pair $w = \{v_i, v_j\}$; N_{L} is the amount of different models contained in the training set.

1.4 Motion information

Motion information is exploited to extract the object from background automatically; i.e., when the user is waving the object in front of the robot, the robot can use the relationship over consecutive frames to discriminate which features belong to the object and which do not. To this end, we first implement the frame difference technique to obtain the region of interest (ROI) on every frame as the approximate position of the target. Only features in this region are extracted and represented in a scheme of co-occurrence visual words. Then we use a k-steps tracking mechanism to build the target model progressively. Every frame in the video sequence is treated as the beginning of a new tracking. The set of pairwise visual words from current image is saved into a buffer. While the next frame is coming, the new set is compared to this buffer and only re-appearing visual words are retained. This mechanism works over every k consecutive frames iteratively as illustrated in Fig. 2. Here the length of every track is 4. The box with a number represents a frame of the image.

At the k-th frame in a track, we treat the surviving features as a graph $G = \langle S, E \rangle$ where S is the set of vertices containing the visual words corresponding to keypoints and $E \subseteq S \times S$ is the set of edges defined by co-occurrence between visual words. The biggest connected component of this graph is found and the visual word pairs of the component are used to create matrix $\boldsymbol{C}$ as described in Section 1.3.

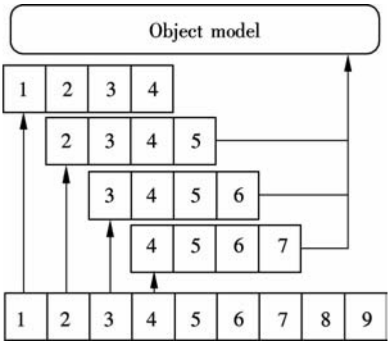

Fig. 2 A flow chart of tracking mechanism

2 Object Recognition

The main task of object recognition in our work is to make decisions on which object appears in the current testing image. First, we extract the features on the entire region of the current testing image and build the co-occurrence graph with the same method described in Section 1. 3. Here we do not follow the tracking mechanism over frames, thus no feature is pruned. All the information for recognition comes from the current frame. We compare two kinds of object recognition methods: the probabilistic method and the logistic regression method.

2.1 Probabilistic model

Given the pairwise visual words vector $\boldsymbol{W} = \{w_1, w_2, \cdots, w_{N_{\mathrm{w}}}\}$, we want to compute the most probable object occurrence in the scene. Using the Bayes rule, the most probable object can be computed as

$$\hat{L} = \arg\max_{L} P(L \mid \boldsymbol{W}) = \arg\max_{L} \frac{P(\boldsymbol{W} \mid L) P(L)}{P(\boldsymbol{W})} \tag{3}$$

where L is the object number; $P(\boldsymbol{W})$ is a normalization that can be omitted when finding the maximum; $P(L)$ is assumed as a uniform distribution. The likelihood function $P(\boldsymbol{W} \mid L)$ is computed for every L by a fast voting method according to the co-occurrence matrix $\boldsymbol{C}$. For every pair w_i presented in the image, we vote for different objects using the occurrence probability of the pair given the object taken from $\boldsymbol{C}$. For the features corresponding to the object with the biggest probability, we find the biggest connected component as the position of the object. At the end of every inference, a confidence of the decision is also given as

$$\phi = \frac{P(L_{\mathrm{w}} \mid \boldsymbol{W}) - P(L_{\mathrm{s}} \mid \boldsymbol{W})}{\sum_i P(L_i \mid \boldsymbol{W})} \tag{4}$$

where ϕ is the confidence of the current inference; L_{w} is the winner of the decision and L_{s} is the object with the second highest score.

2.2 JointBoost

JointBoost was proposed by Torralba et al[11]. Based on the GentleBoost framework[12], it extends boosting methods from the binary classification problem into a multiclass case in an efficient way. Given pairwised feature vectors and labels ($\boldsymbol{v}, z^c$) ($z_i^c = 1$ if example i has ground truth class c, -1 otherwise) as training examples, JointBoost sequentially selects discriminative features to fit the additive model:

$$H(\boldsymbol{v},c) = \sum_{m=1}^{M} h_m(\boldsymbol{v},c) \tag{5}$$

where M is the number of boosting rounds; $h_m(\boldsymbol{v},c)$ are the weak learners and $H(\boldsymbol{v},c)$ is the strong learner. At the current iteration, JointBoost searches a weak learner $h_m(\boldsymbol{v},c)$ shared between classes to optimize the cost function on the training set:

$$J_{\text{wse}} = \sum_{c=1}^{C} \sum_{i=1}^{N} \mathrm{e}^{-z_i^c H(\mathrm{v}_i,c)} (z_i^c - h_m(\boldsymbol{v}_i,c))^2 \tag{6}$$

where N is the number of examples. The cost function J_{wse} can be thought of as an upper bound of the misclassification rate of training examples. To minimize J_{wse}, we adopt the decision stump which is commonly used in the JointBoost framework:

$$h(\boldsymbol{v},c) = \begin{cases} a_S & \text{if } \boldsymbol{v}_i^{\mathrm{f}} > \theta \text{ and } c \in S(n) \\ b_S & \text{if } \boldsymbol{v}_i^{\mathrm{f}} \leqslant \theta \text{ and } c \in S(n) \\ k_S^c & \text{if } c \notin S(n) \end{cases} \tag{7}$$

For those classes sharing this feature ($c \in S(n)$), the decision stump returns $h(v,c)$ depending on the comparison of $\boldsymbol{v}_i^{\mathrm{f}}$ to a threshold θ. For classes not sharing the feature ($c \notin S(n)$), the constant k_S^c prevents the learning procedure from being adversely affected by imbalance between negative and positive training examples. At the end of the current iteration, the training examples are re-weighted as

$$w_i^c := w_i^c \mathrm{e}^{-z_i^c h_m^{S^*}(\boldsymbol{v}_i,c)} \tag{8}$$

to reflect the variance of classification accuracy. When the training is finished, a series of decision stumps are stored sequentially as the strong learner $H(v,c)$ for classifying new samples.

3 Experiments

In this section, we optimize the parameters of the superpixel method and test our proposed system with a series of realistic object detection and recognition tasks. In the experiment, the step S_{t} for searching superpixel seeds is fixed to $S_{\mathrm{t}} = 5$ pixel. We compute the descriptor on a region that is four times greater than superpixel size. The proportion p in Eq. (1) is chosen as $p = 0.87$ for following experiments.

3.1 Length of track

As demonstrated in Fig. 3, the length of every track in our tracking mechanism has an effect on the proportion of features which contribute to the model. A shorter track will give us a model with more noise from the background while a longer track will consume more time on the calculation at every frame. On the other hand, a model with too sparse features also harms the recognition performance. So in our experiments, we manually label 20 images from six different objects (box a, newspapers, tele-control a, toolbox, pad and postcard) respectively as the ground truth data. The positions and regions of the target from 120 images are tagged to test the influence of the length. We examine the correct detection rate and the false alarm rate with different lengths and draw the receiver operating characteristic (ROC) curve in Fig. 4. Then we choose $k = 3$ as the length of the track since this parameter provides a relatively low false alarm rate with an acceptable correct detection rate.

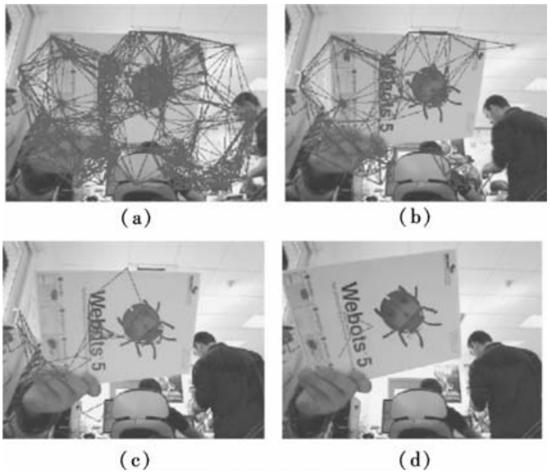

Fig. 3 An example of tracking method with $k = 4$. (a) The first frame; (b) The second frame; (c) The third frame; (d) The fourth frame

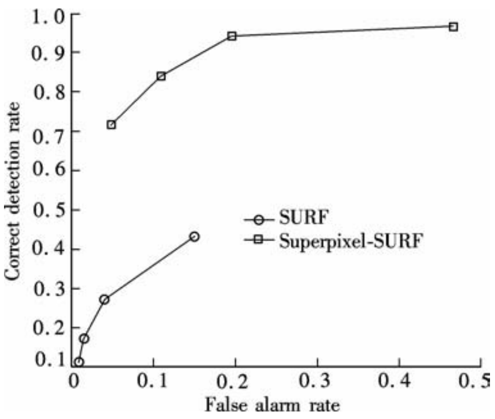

Fig. 4 ROC curve with different lengths of track (k varies from 4 to 1 from left to right)

3.2 Object recognition

We organize groups of experiments on 12 different daily objects: book, magazine, measuring tape, box b, tele-control a, compact disc, box a, toolbox, newspapers, pad, postcard and tele-control b. For each object, a video sequence of 50 images is captured for training matrix $\boldsymbol{C}$. Here the length of the track is $k=3$. In the inference stage, we test 20 images on every object to examine the performance of our method. All the images used in our experiment have a size of 320×240 pixels.

Figs. 5(a) to (d) demonstrate the results of the probabilistic model while the performances of boosted decision stumps are compared in Figs. 6(a) and (b). Fig. 5(a) illustrates the result of only the SURF method over all 12 objects. Apparently the result is unacceptable, especially on those consistently colored targets such as toolbox, compact disc, etc. Thus we adopt the saliency system to automatically try the superpixel-SURF method as the dense sampling strategy for a more abundant model on such objects. In our experiment, the system chooses four objects (tele-control b, compact disc, toolbox and pad) according to an empirical threshold of the saliency measure. The superpixel-SURF method is implemented over these objects to obtain models. In the inference stage, the confidence of each detection is calculated by Eq. (4). When the confidence is too low by using the SURF model, the system reruns the voting method by using the superpixel-SURF model. The result on the same 12 objects is shown in Fig. 5(b).

We also compare the performance of our system under different sizes of the dictionary and illustrate the results in Figs. 5(b), (c) and (d). As one can see, all the results by using the proposed hybrid system are much better than those by using only SURF. Furthermore, as the size of the dictionary increases, the performance is improved slightly.

Finally, we compare the JointBoost method in Figs. 6 (a) and (b). One can see that the JointBoost method outperforms the probabilistic model under different sizes of dictionaries. Although 12 different classifiers are required for obtaining the results on every frame, the computation is still accepted since most of the weak classifiers are efficiently shared over objects. Thus the increase in inference time can be ignored. Also note that the improvement of performance from the larger dictionary is not as obvious as that of the probabilistic model. This is probably because the result is already saturated on the dataset.

4 Conclusion

The system proposed in this paper utilizes the motion information over frames to memorize the shown objects and recognize them in new scenes. The training process of the method only needs very simple interaction from users and it is almost transparent to non-expert users. We can reach an average recognition rate of 98% on a 12-object dataset. In particular, the system can automatically adapt to the object texture level in order to recognize

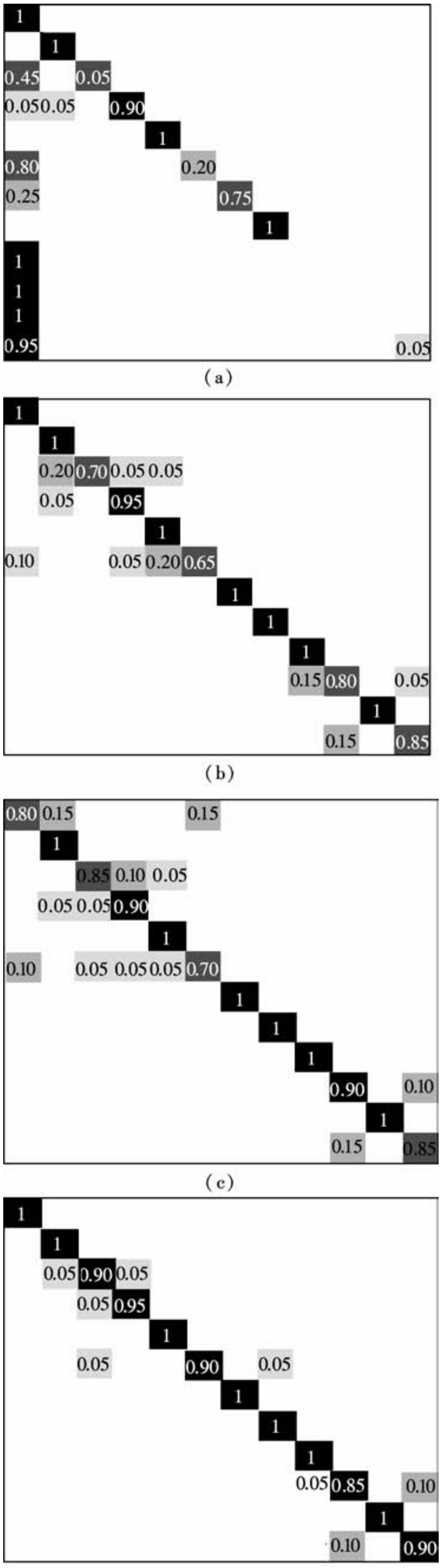

Fig. 5 Confusion matrices of recognition by probabilistic model over 12 objects. (a) SURF only, 1 024 words; (b) Hybrid method, 625 words; (c) Hybrid method, 1 024 words; (d) Hybrid method, 3 125 words

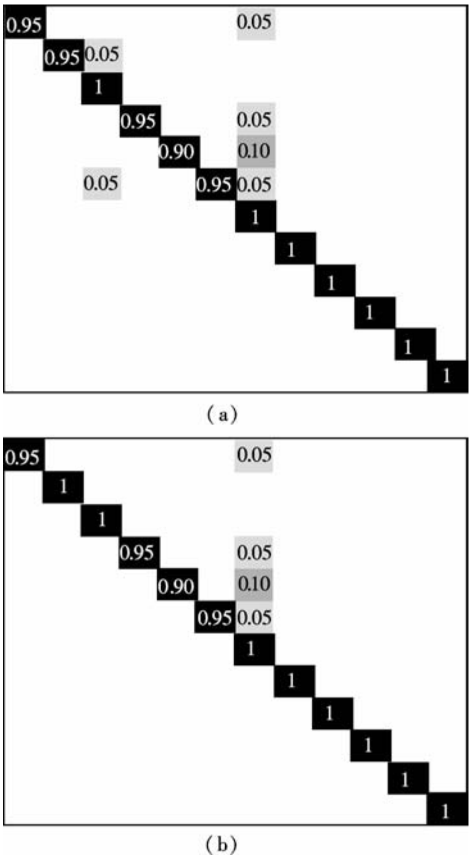

Fig. 6 Confusion matrices of recognition by JointBoost method over 12 objects. (a) Hybrid method, 1 024 words; (b) Hybrid method, 3 125 words

textured as well as textureless colored objects. Here we organize the experiments over a dozen different daily objects because we aim to implement an online learning system on the robot and to encourage the interaction learning between the human and the machine.

References

[1] Sivic J, Zisserman A. Video Google: a text retrieval approach to object matching in videos [C]//*IEEE International Conference on Computer Vision*. Nice, France, 2003: 1470 - 1477.

[2] Lowe D. Distinctive image features from scale-invariant keypoints [J]. *International Journal of Computer Vision*, 2004, **60**(2): 91 - 110.

[3] Bay H, Tuytelaars T, Van Gool L. Surf: speeded up robust features [C]//*European Conference on Computer Vision*. Graz, Austria, 2006: 404 - 417.

[4] Micusik B, Kosecka J. Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry [C]//*IEEE 12th International Conference on Computer Vision Workshops*. Kyoto, Japan, 2009: 625 - 632.

[5] Nowak E, Jurie F, Triggs B. Sampling strategies for bag-of-features image classification [C]//*European Conference on Computer Vision*. Graz, Austria, 2005: 490 - 503.

[6] Filliat D. Interactive learning of visual topological navigation [C]//*IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nice, France, 2008:248 - 254.

[7] Filliat D. A visual bag of words method for interactive qualitative localization and mapping [C]//*IEEE International Conference on Robotics and Automation*. Roma, Italy, 2007: 3921 - 3926.

[8] Jurie F, Triggs B. Creating efficient codebooks for visual recognition [C]//*IEEE International Conference on Computer Vision*. Beijing, China, 2005: 604 - 610.

[9] Nister D, Stewenius H. Scalable recognition with a vocabulary tree [C]//*IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. New York, USA, 2006: 2161 - 2168.

[10] Parikh D, Zitnick C, Chen T. Determining patch saliency using low-level context [C]//*European Conference on Computer Vision*. Marseille, France, 2008: 446 - 459.

[11] Torralba A, Murphy K, Freeman W. Sharing visual features for multiclass and multiview object detection [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007, **29**(5): 854 - 869.

[12] Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting [J]. *The Annals of Statistics*, 2000, **28**(2): 337 - 374.

基于运动信息和局部特征并发性的在线目标检测和识别

张索非[1]　　Filliat David[2]　　吴镇扬[1]

([1] 东南大学水声信号处理教育部重点实验室,南京 210096)
([2]UEI, ENSTA ParisTech, Paris 91762, France)

摘要:实现了一个为类人型机器人设计的目标学习和识别系统,机器人可以利用该系统仅通过和非专业用户简单的互动来发现并记住目标. 当目标展示时,系统利用连续帧间的运动信息提取目标特征并基于视觉单词包方法实现机器学习. 在目标模型的学习与测试阶段,不仅直接使用了局部特征描述子,还使用了局部特征的并发性以提升特征的可鉴别性. 同时,针对目标视觉特征的纹理程度,还采用了一种混合的采样策略. 该混合策略使用了更小的计算资源开销并在一个12 类常见目标构成的集合上取得了良好的识别效果.

关键词:目标识别;在线学习;动作信息;机器视觉

中图分类号:TP391.4