The Fifteenth International Symposium on Artificial Life and Robotics 2010 (AROB 15th '10),
B-Con Plaza, Beppu,Oita, Japan, February 4-6, 2010

Object Recognition Algorithm using Vocabulary Tree and Pre-Matching Array

Ho-Yong Seo and Ju-Jang Lee

Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, Republic of Korea
(Tel : 82-42-350-8032; Fax : 82-42-350-5432)
(E-mail: hyseo@kaist.ac.kr)

Abstract: Vocabulary Tree algorithm builds a tree structure by performing the off-line learning method using the large number of training image datasets. After constructing, we can retrieve a query image class very quickly by searching the tree structure. Because of the great improvement for computation time reduction, this algorithm comes into the spotlight recently.

In this paper, we suggest a method which improves the classification accuracy via searching our tree with multiple times per one test data. The information which given by pre-matching array determines how the tree is visited. Taking our new algorithm, we can reduce miss-classification rate considerably. On the other hand, losses from computation time and memory allocation are negligible.

Keywords: Vocabulary Tree, Pre-Matching Array, Object Recognition

I. INTRODUCTION

Object recognition field which is a one of central parts in computer vision, deals with the method that determines the unknown class of test image from given training image datasets.

When constructing training image datasets from a large number of images, we can't use whole image pixel information for computation time and memory allocation problems. Recently, the method, so called keypoint based feature extraction technique, is widely used for making training data values. One of SIFT, SURF, MSER, CenSurE and the others is used for each proper design goal. These techniques use some meaningful pixels which appear more frequently than other pixels when image information such as scale, orientation, illumination is changed. Using above feature extraction technique, we can get D-dimensional vector for each keypoint. These vectors are used for our training image datasets [1].

When Object recognition based on above feature extraction methods performs learning process, we can categorize object recognition by which learning method is used among various kind of method such as offline learning, online learning, supervised learning and unsupervised learning.

Offline learning is done before test input is coming. In other words, when test step is on progress, there isn't additional learning. On the other hand, online learning performs learning step and test step simultaneously. Supervised learning is used when we know all classes of training image. Otherwise, we can use unsupervised learning to categorize all images [2].

The most general method used for object recognition is machine learning techniques such as support vector machine(SVM), kernel machine, expectation-maximization(EM) and their variations [2]. These techniques assume that the labels of our training image datasets are already known. In other words, supervised learning case is popular for object recognition field.

But vocabulary tree case, which is mainly referred by our paper, uses K-means clustering and tree structured vocabulary. The former allow us using unlabeled training datasets easily [3], and the latter give less computation time with respect to other method which doesn't use tree structure. And, the retrieval performance is about 70~80% [4].

In this paper, we tried to get more accurate retrieval performance than that of previous vocabulary tree case. Once the tree is constructed, when performing test process through this tree, we are just required very short computation time. So, we try to search tree structure in multiple times per one test image for the object of improvement of performance. Because we use tree structure, multiple time searching doesn't need too much additional calculation time. Also, by adding simple array so called pre-matching array, we can do multiple tree searching process efficiently just using few additional memory space.

II. REVIEW OF VOCABULARY TREE

1. Characteristics

Using vocabulary tree(VT) is usable whether training datasets have label or not. VT always categorize class of test data very quickly with reasonable performance.

At training step, we construct a tree using K-means clustering which is one of popular methods in unsupervised learning. At test step, we try to search that tree, then our computation time is expressed with $O(KLD)$ where K is branch factor of tree, L is depth of tree and D is input vector dimensions as described Fig.1 [4].

2. Implementation

A. Training Step

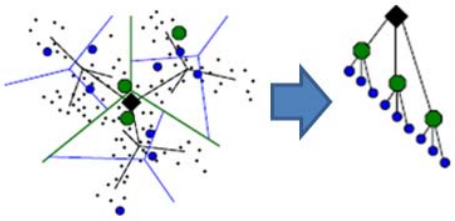

Fig.1 Tree construction Process when K=3,L=2 [4]

Step1. Using one of keypoint feature extraction algorithm, extract all feature D-dimensional vectors from all training image sets.
Step2. Define some constant K. And, categorize features for K-children node using K-means clustering based on all feature vectors within current node.
Step3. Define tree depth constant L. Repeat Step 2 until arrives at leaf-node.
Step4. At each leaf node, save number of training image numbers(N_i) within ith node.
Step5. Save the number of feature vectors(n_i) for each image within ith leaf node. Repeat for all images within ith leaf node.

B. Test Step

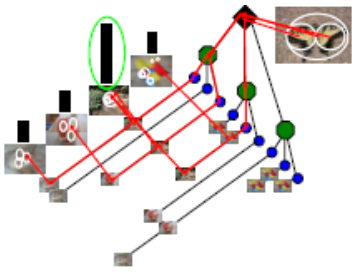

Fig.2 Testing Process concept [4]. Minimum scoring described as maximum counting.

Step1. Extract all feature vectors from test image.
Step2. Start at root node.
Step3. For each feature vector, compute Euclidean distance with this vector and K-means vector.
Step4. Go child node which give smallest distance number. Repeat until arrive at leaf node.
Step5. Repeat Step 3-4 for all feature vectors from test image.
Step6. Count and save the number of feature vectors(m_i) within ith leaf node. Repeat for all leaf node.
Step7. For each training image, calculate score s as follows.

$$w_i = \ln \frac{N}{N_i} \tag{1}$$

where N is total number of training image.

$$\begin{aligned} q_i &= n_i w_i \\ d_i &= m_i w_i \end{aligned} \tag{2}$$

$$s = \left\| \frac{q}{\|q\|} - \frac{d}{\|d\|} \right\|_2^2 = 2 - 2 \sum_{i|q_i \neq 0, d_i \neq 0} q_i d_i \tag{3}$$

Step8. For all training image, find minimum score. And, return image index information which give minimum score value as Fig.2.

These all steps are progressed by referring [4].

III. PRE-MATCHING ARRAY

1. Characteristics

If we change the index returning method that retrieve not only one image but also multiple images, then we expect that test process give correct classification for all images returned. But, as we mentioned, average retrieval performance is about 70~80%, in other words, not perfect.

So, we suggest pre-matching array for the object of improvement of retrieval performance, maintaining fast computation time as possible using VT structure.

2. Definition

Pre-matching array contains retrieved training image indexes with high scored sequence from one test image. So, array size(N_p) determines how many images we consider.

And regarding each saved training image as new test image, we can search tree structure again and get another result image indexes.

Next, among these all result image indexes, we have to extract image indexes regard as same class with our test image. This process is performed by using new variable defined R .

Here, R is defined as the number of count that calculated from all result image indexes. So, from $N_p(N_p - 1) \times 1$ vector R , we can get best image index what we want to extract. When using training image as test image, we have to eliminate first resulting image index because first index is always that of training image itself.

3. Implementation

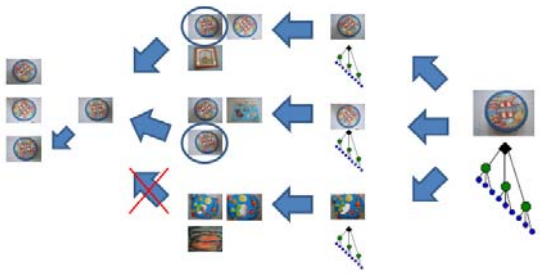

Fig.3 Correction of miss-classification by using pre-matching array

Step1. Construct VT using training image dataset.
Step2. From putting test image into VT, get N_p number of training image indexes which has best score sequentially.
Step3. For all selected training images, put those images into VT, get $N_p\left(N_p-1\right)$ number of result image indexes except of first result image index for each selected training image.
Step4. Count for each result image index, and save that result at R.
Step5. Return one result image index which has best R count and replace lowest counted training image index as our selected image index as Fig.3.

IV. EXPERIMENT & RESULT

1. Parameter Setting

Among wide variety of keypoint feature extraction method, we select SURF algorithm for the object of fast calculation via filter scaling method [1]. We utilized SURF inside OpenCV library.

When performing tree construction process, we have to determine K and L. As referred from [4], to get reasonable performance, we select K=10 and L=6.

Next, we have to determine training image contents and numbers. As Fig.4, we select same training images with [4]. Four images exist for each one specific object.

Fig.4. Part of our training image dataset [4].
Total number of training images is 1000, in other words, 250 objects. Our purpose is relative comparison with previous VT and our pre-matching array algorithm, for simplicity, number of training images is fixed.

2. Result

A. Basic training image dataset case

Using above training image dataset, we extract feature vectors from SURF, construct tree and classifying randomly selected test images. Our result values are tree construction time, test time, and performance for all randomly selected test images.

Table 1. parameter variation for Np when fixed training image dataset is used.

	Construction time	Test time
0(VT)	356.8sec	39.3sec
1	356.9sec	40.8sec
2	355.3sec	40.3sec
3	352.8sec	41.5sec
4	357.7sec	41.5sec
5	358.0sec	41.3sec
6	360.2sec	42.2sec
7	356.1sec	42.7sec
8	355.6sec	43.6sec
9	358.3sec	42.9sec

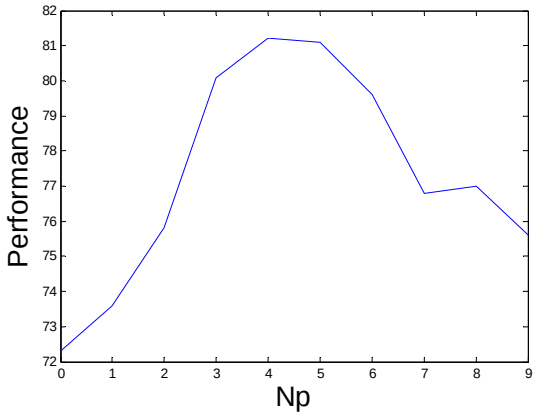

Fig.5. Performance graph when fixed training image dataset is used.(Best at Np=4)

As we expected, computation times for both construction and test case are slightly increase with respect to that of basic VT case. But, those increment value is negligible.

And, when Np=4, best performance is returned and this value is much better than that of basic VT case. Best performance is taken when Np equals 4. Because we use training images which have equally 4-images for each one class, Np=4 give us best performance.

B. Random selected training image dataset case

To verify generalization which means our algorithm is robust for changing training image datasets. So, in this step, we select randomly 1000 images regardless of equal number of class for equal number of image groups.

Our result is as follows.

Table 2. parameter variation for Np when randomized training image dataset is used.

	Construction time	Test time
0(VT)	358.3sec	40.6sec
1	358.1sec	40.9sec
2	357.7sec	40.5sec
3	359.8sec	41.1sec
4	362.3sec	41.0sec
5	366.8sec	40.3sec
6	361.5sec	49.9sec
7	353.7sec	40.2sec
8	361.1sec	42.5sec
9	359.6sec	41.8sec

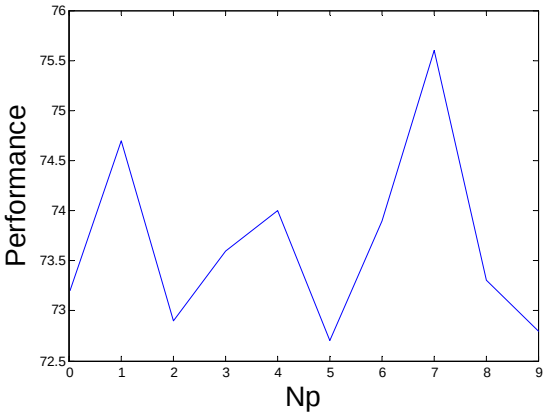

Fig.5. Performance graph when randomized training image dataset is used.

If our data is selected randomly, the Np value which give us best performance can be changed. Also, Our best performance value is decreased. Np = 5 case give us worse than that of basic VT tree case. For generalization, the method that determines good Np value have to be studied later.

Of course, computation time is very close to constant.

V. CONCLUSION

Combining vocabulary tree and pre-matching array, we can get considerably improved retrieval performance with compensating additional small memory allocation and computation time. The tree structure gives us almost same calculation time. And, just additional $N_p\left(N_p-1\right)\times 1$ memory allocation is needed. Our algorithm can be applied for auto categorization for unlabeled digital camera pictures or unlabeled image at web search engine which requires high accuracy for retrieval task.

If the number of images in one class is varying greatly, in other words, when we use random unlabeled image, another N_p selection method have to be introduced. We tried this problem later for generalization.

VI. ACKNOWLEDGEMENT

Authors are gratefully acknowledging the financial support by Agency for Defence Development and by UTRC(Unmanned Technology Research Center), Korea Advanced Institute of Science and Technology.

REFERENCES

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool (2006), Speeded-Up Robust Features(SURF). Springer 3951 : 404-417

[2] Christopher M.Bishop(2006), Pattern Recognition and Machine Learning. Springer : 1-4

[3] J.Sivic , A.Zisserman(2003), Video Google : A Text Retrieval Approach to Object Matching in Videos. Citeseer 2: 1470-1477

[4] D.Nister, H.Stewenius(2006), Scalable Recognition with a Vocabulary Tree. Citeseer 5