COMPACT VISUAL CODEBOOK FOR ACTION RECOGNITION

Qingdi Wei[1]*, Xiaoqin Zhang*[1]*, Yu Kong*[2]*, Weiming Hu*[1]*, and Haibin Ling*[3]

[1]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2]Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing, P.R. China
[3]Computer & Information Science Department, Temple University, Philadelphia, PA,USA
{qdwei,xqzhang,wmhu}@nlpr.ia.ac.cn, kongyu@bit.edu.cn, hbling@temple.edu

ABSTRACT

Visual codebook has been popular in object classification as well as action analysis. However, its performance is often sensitive to the codebook size that is usually predefined. Moreover, the codebook generated by unsupervised methods, e.g., K-means, often suffers from the problem of ambiguity and weak efficiency. In other words, the visual codebook contains a lot of noisy and/or ambiguous words. In this paper, we propose a novel method to address these issues by constructing a compact but effective visual codebook using sparse reconstruction. Given a large codebook generated by K-means, we reformulate it in a sparse manner, and learn the weight of each word in the original visual codebook. Since the weights are sparse, they naturally introduce a new compact codebook. We apply this compact codebook to action recognition tasks and verify it on the widely used Weizmann action database. The experimental results show clearly the benefits of the proposed solution.

Index Terms— Codebook, sparse representation, action recognition

1. INTRODUCTION

Bag of visual words has become an important framework in many computer vision areas, due to their robustness and simplicity, as well as reported excellent performance on visual recognition tasks such as object recognition, scene identification, and action understanding [1, 2, 3, 4]. The basic idea is to construct a visual codebook on the statistics of the various features in videos and images, which can be used to represent a target. In recent years, the K-means algorithm has been frequently used to construct the visual codebook because of its effectiveness and simplicity. The centroid of a cluster corresponds to a word in the visual codebook. However, K-means has two major shortcomings. First, we need to set the number of clusters in advance, which has a decisive role in the performance in the consequent recognition tasks. Second, there is no guarantee that every word from K-means carries *a semantic meaning*, in that it is a purely unsupervised method. Thus the resulting visual codebook often contains a lot of noisy and ambiguous words.

In this paper, we add an additional layer of compression after the traditional codebook construction to overcome these drawbacks, as shown in Fig. 1. First, we generate a sparse representation of the training data based on the old visual codebook. Second, we learn the weight of each word in the old visual codebook from the sparse representation. At last the compression can be done based on the weights.

We test the proposed approach on the Weizmann action database [5]. Interestingly, the new visual codebook, half the size of the old visual codebook, has the same performance as the one built by K-means .

The rest of the paper is organized as follows: in the next subsection we discuss the related work, and the sparse representation method is introduction in Section 2. The action feature extraction and recognition methods are presented in Section 3, and the experimental results are shown in Section 4. Section 5 is devoted to conclusion.

1.1. Related work

To optimize the initial visual codebook, many approaches have been proposed in the literature. For instance, Farquhar *et al*. built their codebook based on maximum-a-posterior [6]. Moosmann *et al*. constructed visual codebooks using randomized clustering forests [7]. Winn *et al*. learned an optimally compact visual dictionary by pair-wise merging of visual words from an initially large dictionary [3]. Yang *et al*. unified the visual codebook generation with classifier training [1]. The algorithms mentioned above are used to generate a discriminative codebook. In this paper, we focus on reducing the size of the visual codebook but without compromising performance.

Another piece of closely related area is sparse representation in computer vision. It has gained much attention in recent

Thanks to the NSFC (Grant No. 60825204 and 60935002) and the 863 High-Tech R&D Program of China (Grant No. 2009AA01Z318) for funding. H. Ling is supported in part by NSF Grant No. IIS-0916624.

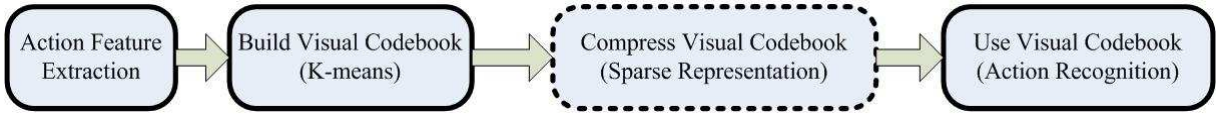

Fig. 1. Flowchart of the usage of visual codebook (Dashed part is additional)

years due to its compact and robust representation power and it has been widely applied in many computer vision tasks, such as face recognition [8], tracking [9], background subtraction [10], texture segmentation [11], lighting estimation [12], etc.

2. SPARSE REPRESENTATION OF ACTION

When we use the space-time interest point [13] as the motion feature, there are many interest points that are extremely similar, because of the periodic nature of human motion and the similarity of different people doing the same motion. Therefore, a feature point (word) usually can be represented by a linear combination of some typical points (words). Given a visual codebook $V = [v_1, v_2, \cdots, v_n] \in \mathbb{R}^{d \times n}$, containing n motion words $v_i \in \mathbb{R}^d$, a new motion word t can be approximated by a linear combination of the words,

$$t \approx Vx = x_1 v_1 + x_2 v_2 + \cdots + x_n v_n, \tag{1}$$

where $x = (x_1, x_2, \cdots, x_n)^T$ is the coefficient vector. When the codebook is big enough, $x = [0, \cdots, 0, \beta, \cdots, 0, \cdots, 0]^T$ is a sparse vector, forming the sparse representation of t on V. We can find the sparsest solution to $t = Vx$, by solving the following optimization problem:

$$x = \arg\min ||x||_0 \text{ subject to } Vx = t, \tag{2}$$

where $||.||_0$ denotes the ℓ_0 norms. This could be essentially treated as an ℓ_1-regularized least squares problem [14].

2.1. Sparse Representation of Codebook

Our goal of constructing codebook is to use fewer words to represent more motions, while consistent with the sparse representation in Eqn. (2). Thus we compress the visual codebook using sparse representation, after the construction of the visual codebook, as shown in the Fig. 1.

The old visual codebook V_{old} is built by K-means, then according to Eqn. (1), the new target t can be represented by

$$V_{old} x_{old} = t. \tag{3}$$

Our goal is to build a compact visual codebook

$$V_{new} = V_{old} S, \tag{4}$$

where $V_{new} = [v_1, v_2, \cdots, v_m] \in \mathbb{R}^{d \times m} (m < n)$ is the new codebook which contains fewer words than the old one. Then Eqn. (3) can be reformulated as

$$V_{new} x_{new} = t. \tag{5}$$

Ideally, x_{old} is a sparse vector. $x_{old} = x_{new}$, when we add many 0 to x_{new} to make it have the same dimension as x_{old} does. Meanwhile, S is a diagonal matrix, with 0 and 1 on its main diagonal. S satisfies $V_{old} S x_{new} = t$, so the ideal codebook for t is $V_{new} = V_{old} S$. Therefore, we can use a large number of t to get a compact V_{new}.

In another word, x is the sparse representation of t on V in Eqn. (1). The non-zero element in x, β, means that its corresponding word is used to represent t. So, it is one of the words we need.

The first step of our approach to compressing the old visual book is to solve

$$\min ||V_{old} sr_i - feature_i||_2^2 + \lambda ||sr_i||_1, \tag{6}$$

where $feature_i$ is one of the action features, whose corresponding sparse representation is $sr_i \in \mathbb{R}^n$. It means we use the old visual codebook to sparsely represent the training data following Eqn. (6) that is solvable via an interior-point method[1] [14].

The second step of our approach is to learn the weight of each word in the old visual codebook, from the sparse representation of the entire action features. Then we generate the new visual codebook through six different algorithms. The first two are extensions of SRC, whose details can be found in [8]. For $sr_i \in \mathbb{R}^n$, $\delta_j(sr_i)$ is a new vector whose only non-zero element is the jth element in sr_i. Then we can approximate the given test sample $feature_i$ as $\overline{feature_i} = V_{old} \delta_j(sr_i)$. We then identify the major words in the linear combination of $feature_i$ by Eqn. (7). Here $(.)_j (j \in [1, \cdots, d])$ denotes the value on the jth dimension in (.).

$$\arg\min_j ||feature_i - V_{old} \delta_j(sr_i)||_2 \tag{7}$$

Algorithms 3 and 4 choose the biggest coefficients associated with the corresponding word as the major word in the linear combination of $feature_i$. The last two algorithms count several largest values that are similar as choosing eigenvectors in PCA.

The six algorithms can be divided into two categories. One focuses on ℓ_1 distance and the other uses ℓ_0 distance. The inputs of the six algorithms are V_{old}, sr, and $feature$, while the output is $W \in \mathbb{R}^d$. W are the weights of words in the old visual codebook. The bigger values in W correspond to the more important words. In this paper, a new visual codebook is built from W through Algorithm 7, where α is a parameter controlling the compression rate. When $\alpha = 1$, the new codebook is same as the old codebook.

[1] http://www.stanford.edu/ boyd/l1_ls/

for $i = 1; i \leq total$ **do**
  $r(feature_i) = ||feature_i - V_{old}\delta_j(sr_i)||_2$;
  $temp = \arg\min_j r(feature_i)$;
  $w_{temp} = w_{temp} + ||(sr_i)_{temp}||_1$;
end

Algorithm 1: SRC+ℓ_1

for $i = 1; i \leq total$ **do**
  $r(feature_i) = ||feature_i - V_{old}\delta_j(sr_i)||_2$;
  $temp = \arg\min_j r(feature_i)$;
  $w_{temp} = w_{temp} + ||(sr_i)_{temp}||_0$;
end

Algorithm 2: SRC+ℓ_0

for $i = 1; i \leq total$ **do**
  $temp = \arg\max_j (sr_i)_j$;
  $w_{temp} = w_{temp} + ||(sr_i)_{temp}||_1$;
end

Algorithm 3: MAX+ℓ_1

for $i = 1; i \leq total$ **do**
  $temp = \arg\max_j (sr_i)_j$;
  $w_{temp} = w_{temp} + ||(sr_i)_{temp}||_0$;
end

Algorithm 4: MAX+ℓ_0

for $i = 1; i \leq total$ **do**
  $[value, index] = sort((sr_i)_j, 'descend')$;
  $\frac{\sum_{k=1}^{temp}(value)_k}{\sum_{j=1}^{n}(value)_j} \geqslant 90\%$;
  $w_{index(1:temp)} = w_{index(1:temp)} + ||(sr_i)_{index(1:temp)}||_1$;
end

Algorithm 5: MAXS+ℓ_1

3. ACTION RECOGNITION

Our main framework for action recognition follows [15]. In the training stage, we build the old visual codebook using K-means. Then we generate the new visual codebook based on sparse representation and learn the probability table from the new visual words to action classes. In the testing stage, we class the action features to words in the new visual codebook and sum the probabilities from word to action of all action features in one video which can be found in the probability table. The largest value leads to our prediction of the action class.

for $i = 1; i \leq total$ **do**
  $[value, index] = sort((sr_i)_j, 'descend')$;
  $\frac{\sum_{k=1}^{temp}(value)_k}{\sum_{j=1}^{n}(value)_j} \geqslant 90\%$;
  $w_{index(1:temp)} = w_{index(1:temp)} + ||(sr_i)_{index(1:temp)}||_0$;
end

Algorithm 6: MAXS+ℓ_0

Input: V_{old}, W, α
for $i = 1; i \leq total$ **do**
  $[value, index] = sort(W, 'descend')$;
  $\frac{\sum_{k=1}^{temp} W_k}{\sum_{j=1}^{n} W_j} \geqslant \alpha$;
  choose the largest $temp$ weights corresponding to words
end
Output: V_{new}

Algorithm 7: Construct New Codebook

The space-time interest point is used for a compact representation of video data and robust to occlusions, background clutter, significant scale changes, and high action irregularities. They have been successfully used as action features in the action recognition tasks [2, 16]. In our work, we detect space-time interest points using the Harris operator detector, with hog+hof as descriptors, using a publicly available code.[2] The Euclidean distance metric is adopted to evaluate the similarity of two features.

4. EXPERIMENTS

We evaluate our approach on the Weizmann action database [5]. It contains 81 low-resolution video sequences from nine different people. Each sequence shows one person performing one of the nine natural actions: "running", "walking", "jumping", "jumping forward on two legs", "jumping in place on two legs", "galloping sideways", "waving two hands", "waving one hand", and "bending". The video sequences have 180×144 pixel resolution and 25 frames per second. We perform the leave-one-out-cross-validation (LOOCV), which means 80 videos are used for training and the rest one for testing.

We first test our algorithm with different values of α and different numbers of words in the old visual codebook. The results are shown in Table 1. We can see that the more words

[2] http://www.irisa.fr/vista/Equipe/People/Laptev/download.html

Table 1. Recognition rate vs. the number of words in the new visual codebook, with different α. Results are the average of six algorithms.

# of words	1000	800	400	200
$\alpha = 1$	90.12%	87.65%	83.95%	75.84%
0.9	780	624	316	161
	89.51%	87.24%	81.89%	71.81%
0.8	637	510	260	133
	89.51%	84.16%	80.25%	71.40%
0.7	518	415	213	109
	89.30%	84.16%	76.75%	66.67%
0.6	414	333	171	88
	84.65%	81.69%	70.37%	59.46%

Table 2. Recognition rate of the six algorithms with different α. Percentages are recognition rate and the integers are the new codebook size.

Alg.	1	2	3	4	5	6
α=1	90.12%	90.12%	90.12%	90.12%	90.12%	90.12%
0.9	775	760	775	761	809	798
	90.12%	90.12%	88.89%	90.12%	88.89%	88.89%
0.8	631	618	631	618	670	657
	88.89%	91.36%	88.89%	91.36%	88.89%	87.65%
0.7	511	**499**	511	500	551	537
	87.65%	**92.59%**	88.89%	91.36%	87.65%	87.65%
0.6	406	398	406	398	445	430
	87.65%	87.65%	86.42%	87.65%	88.89%	87.65%

the old visual codebook has, the better performance our approach achieves.

In Table 2, our old codebook has 1000 words, with a recognition rate of 90.12% (73 out of 81 times). We compress the number of words with different α, and we can see that the number of words drops sharply, while the recognition rate decreases just slightly. Furthermore, when $\alpha = 0.7$, Algorithm 2 gets the highest recognition rate that is two percentage points higher than the old codebook with the original word size, and it only needs 499 words, just half the size of the old codebook. Overall, the six algorithms of building the new visual codebook get very comparable recognition rates (70-75 out of 81 times). It can be explained by that when sr_i is indeed sparse, the results are roughly the same across the 6 algorithms. In another word, the distribution of the data of interest is indeed consistent with our conjecture. Compared with the old codebook containing 1000 words, our new visual codebook is half in size and has better performance.

5. CONCLUSION

In this paper we proposed a method using sparse representation to compress the visual codebook. We first represent the training data using sparse representation of the old visual codebook, and then learn the weight of every word that is used for compression. We tested our approach on the Weizmann action database, and showed that the sparse representation could help optimize the visual codebook learnt by K-means to fewer words while with stable performance.

6. REFERENCES

[1] L. Yang, R. Jin, R. Sukthankar, and F. Jurie, "Unifying discriminative visual codebook generation with classifier training for object category recognition," in *CVPR*, 2008.

[2] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," in *BMVC*, 2006.

[3] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *ICCV*, 2005.

[4] Q. Wei, X. Zhang, Y. Kong, W. Hu, and H. Ling, "Group action recognition using space-time interest points," in *ISVC*, 2009.

[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *PAMI*, 2007.

[6] J. Farquhar, S. Szedmak, H. Meng, and J. Shawe-Taylor, "Improving bag-of-keypoints image categorisation," Technical report, University of Southampton, 2005.

[7] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *NIPS*, 2007.

[8] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Yi Ma, "Robust face recognition via sparse representation," *PAMI*, 2009.

[9] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *ICCV*, 2009.

[10] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *ECCV*, 2008.

[11] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *CVPR*, 2008.

[12] X. Mei, H. Ling, and D.W. Jacobs, "Sparse representation of cast shadows via ℓ_1-regularized least squares," in *ICCV*, 2009.

[13] I. Laptev, "On space-time interest points," *IJCV*, 2005.

[14] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ_1-regularized logistic regression," *JMLR*, 2007.

[15] Q. Wei, W. Hu, X. Zhang, and G. Luo, "Dominant sets-based action recognition using image sequence matching," in *ICIP*, 2007.

[16] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *VS-PETS*, 2005.