Clustering Analysis for Codebook Generation in Action Recognition using BoW Approach

Jordi Bautista-Ballester, Jaume Vergés-Llahí and Domenec Puig

***Abstract*—In computer vision, action recognition is a common topic of the State of the Art. Bag of Visual Words method has been recently widely used for this topic. In this paper, we intend to show how clustering the information extracted from the videos has its influence in the final recognition results. We propose an analysis of three different clustering methods, namely: K-means, Meanshift and using a random selection of the cluster centers. Our work lies on the approach of action recognition through Bag of Visual Words representation and it is motivated by the need of knowing the adequate number of words for representing a set of videos, in the sense that as much words the codebook has, better recognition we reach, but with an increase of computing time. First, Harris interest points have been determined in order to extract a 3D descriptor based on Histogram of Gradients for each of these points. Then, we extract a representative codebook of these descriptors by clustering or selecting them randomly. After quantization, the representation of the videos is done by computing histograms of the information. Finally a Support Vector Machine is used to classify actions in videos. Experimental results are obtained over a public action dataset.**

***Index Terms*—Bag of Words, Action Recognition, K-Means, Codebook generation**

Jordi Bautista-Ballester is with Ateknea Solutions Catalonia and Universitat Rovira i Virgili.
E-mail: jordi.bautista@ateknea.com, estudiants.urv.cat

Jaume Vergés-Llahí is with Ateknea Solutions Catalonia.
E-mail: jaume.verges@ateknea.com

Domenec Puig is with Universitat Rovira i Virgili.
E-mail: domenec.puig@urv.cat

I. Introduction

Action recognition is a very active research topic in computer vision with many important applications, including video surveillance, human computer interaction, robotics, programming by demonstration, among others. Action recognition is the process of naming actions, usually as an action verb. To reach that goal, many approaches typically make use of a combination of vision and machine learning techniques. Vision techniques try to extract action features from the videos, while machine learning techniques try to learn statistical models from those features, and classify new features using the learned model. A wide range of Action Recognition methods exists and they can be classified by spatial or temporal representations.

In our approach we make use of a common approach called spatio-temporal Bag of Words representation. In this method, spatial and temporal information are extracted from the surroundings of an interest point (IP) in order to build the feature descriptor. From this set of features a dictionary is computed and quantized to represent each snippet of the video. To learn a model and further classification of new features, a Support Vector Machine (SVM) is employed.

The point of our work is to show the influence of different methods for clustering information extracted from the image, i.e. to build a good codebook. And also to find why K-means algorithm is widely used for

codebook generation, instead, for example, randomly selecting the centers.

A. Related Work

In the related work, authors who make use of BoW approach, use to vary the three main phases: feature extraction and description, vector quantization and pooling. In [2] an extended Harris corners was employed to detect spatio-temporal interest points (STIP), to compute a descriptor composed by a Histogram of Gradients concatenated with a Histogram of Optical Flow (HoG-HOF descriptor). They used a K-means algorithm to cluster the features and a Nearest Neighbor with euclidean distance for pooling. Finally, an SVM with χ^2 kernel was used to learn a model and classify new instances. A novel approach were proposed in [5], in which they built a Dense grid of interest points and extracted a trajectory based descriptor (Dense trajectories). More recently, Zhang, et.al. [7] made use of sparse code to create the dictionary, a variant of the K-means clustering algorithm where they tuned the constraint with a λ parameter in order to make it less restrictive.

B. Our approach

The main amount of proposals use a supervised clustering for codebook generation. We analyze this procedure, which has been widely seen that it has high influence to the final recognition performance. We know that the recognition performance steadily grows with the size of the codebook, as observed, e.g. by [14]. To this purpose we compare three different methods. Firstly, we use standard K-means [11], which is the most common clustering algorithm for this topic. In this algorithm a K-value is needed to be provided as the final number of words representing all data collected from videos. A principal disadvantage of standard K-means is that clusters can only be separated by a hyper-plane. Using a weighted kernel K-means [8], nonlinear separators can be obtained. Secondly, we propose to use Meanshift [10], in which the final number of clusters is not previously determined. By this, we leave the algorithm to determine itself the codebook size by tuning a bandwidth parameter. As third option, we propose the random selection of the cluster centers. In the end, following BoW procedure, we find the word which represents each snippet and we use a non-linear SVM classifier to classify the videos. Our contribution in this paper is the analysis of data clustering, showing the influence of the dictionary size and comparing the results obtained by three different methods, using the final recognition performance as the evaluation metric.

II. Methodology

The method we propose can be seen in Fig.1. We firstly extract some interest points (IP) from video frames with Harris detector. Secondly, we compute HoG3D descriptors for each IP. To build the codebook, and due to the computing complexity, we limit the amount of data to 100.000 samples, and then it is computed by a clustering algorithm, which gives a dictionary with a specific number of words. We code, then, each snippet by a codebook word. Finally, a Support Vector Machine classifier (SVM) is used to evaluate the performance of the recognition, which is used as the evaluation metric.

We divide this section in three subsections: in A, we explain how vectors are computed, i.e., how we extract descriptors for interest points in video frames. In B, we explain how we obtain the codebook, which will be used for representing the sequence snippets. In this subsection, we introduce K-means, Meanshift and random sample approaches. Finally, in subsection C, we explain how Bag of Words proposal is being used to finally evaluate the recognition performance.

A. Feature extraction

We extract Harris corners as interest points (IP) in a similar way as it is done in [2].

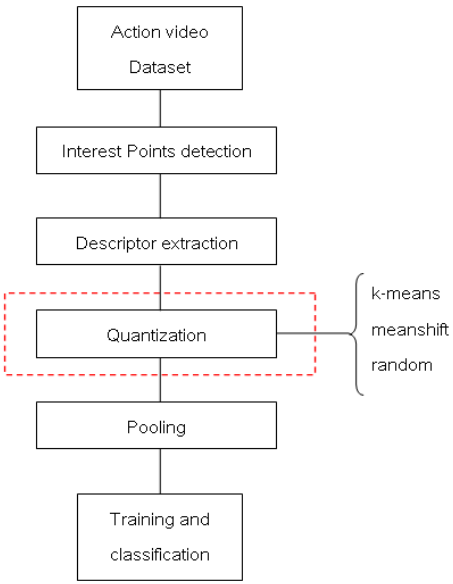

Fig. 1. Flow chart of the methodology used to evaluate the three proposed clustering methods for action recognition.

For a sampling point s, position coordinates (x_s, y_s) are kept to which are added to them the third dimension, which is the time (t_s), i.e., the frame number, and the spatial and temporal scale (σ_s, τ_s), both used to determine at which scale the descriptor is computed. We use HoG3D descriptor [1] to code the characteristics of these IP. Histogram of Gradients (HoG) have been widely used for object recognition in static images, but, in our approach, we need a descriptor which can relate spatial and temporal information. HoG3D is similar to HoG descriptor, in the sense that it computes the gradient histograms of a pixel surroundings, but with the main advantage that it generalizes the concept to 3D. The final descriptor d_s for s is computed for a local support region r_s with a width (w_s), height (h_s) and length (l_s) around the position s, given by $w_s = h_s = \sigma_0 \sigma_s$ and $l_s = \tau_0 \tau_s$, where σ_0 and τ_0 parameters characterize the relative size of the support region around s.

This local support region r_s is divided into a set of $MxMxN$ cells c_i. For each cell, an orientation histogram is computed by $h_c = \sum_{i=1}^{S^3} q_{b_i}$, where q_{b_i} is the quantization employing a regular polyhedron for the subblock b_i. Finally, all histograms are concatenated to one feature vector $d_s = (d_1, ..., d_{M^2N})^T$. Final dimension of the feature can be precalculated by $dim\{d_s\} = M^2 \cdot N \cdot n$, with n the number of orientations, taking into account its full or half orientation. The relevance of this value lies in the computation time when clustering, which considerably increases as higher this dimension is.

B. Codebook generation

Traditionally, in action recognition, the codebook is generated by previously setting the number of words that it is desired to have. In this sense, K-means clustering algorithm gives quite good results. The intention of this work is to show why K-means seems to be better for that purpose than algorithms that can decide the number of clusters by themselves, just tuning some parameters, or even selecting the clusters randomly.

1) K-means: Given a set of vectors, the K-means algorithm seeks to find clusters that minimize the objective function:

$$D(\{\pi_c\}_{c=1}^{k}) = \sum_{c=1}^{k} \sum_{a_i \in \pi_c} \|a_i - m_c\|^2 \quad (1)$$

$$\text{where } m_c = \frac{\sum_{a_i \in \pi_c} a_i}{|\pi_c|}$$

The centroid (or the mean) of the cluster π_c is denoted by m_c. A principal disadvantage of standard K-means is that clusters can only be separated by a hyper-plane.

2) Meanshift: The Meanshift algorithm is a non-parametric clustering technique which does not require prior knowledge of the number of clusters, and does not constraint the shape of the clusters. Given n data points $x_i, i = 1, .., n$ on a $d-$dimensional space R^d, the multivariate kernel density estimate obtained with kernel $K(x)$ and window radius h is:

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \qquad (2)$$

For radially symmetric kernels, it suffices to define the profile of the kernel $k(x)$ satisfying

$$K(x) = c_{k,d} k(\|x\|^2) \qquad (3)$$

where $c_{k,d}$ is a normalization constant which assures $K(x)$ integrates to 1. The modes of the density function are located at the zeros of the gradient function $\nabla f(x) = 0$. The gradient of the density estimator is

$$\nabla f(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) g(\| \frac{x - x_i}{h} \|^2) \qquad (4)$$

where $g(s) = -k'(s)$. The first term is proportional to the density estimate at x computed with kernel $G(x) = c_{g,d} g(\|x\|^2)$ and the second term

$$m_h(x) = \frac{\sum_{i=1}^{n} (x) g(\| \frac{x - x_i}{h} \|^2)}{\sum_{i=1}^{n} g(\| \frac{x - x_i}{h} \|^2)} \qquad (5)$$

is the Meanshift. The Meanshift vector always points towards the direction of the maximum increase in the density. The Meanshift procedure, obtained by successive computation of the Meanshift vector $m_h(x^t)$, and translation of the window $x^{t+1} = x^t + m_h(x^t)$, is guaranteed to converge to a point where the gradient of density function is zero.

The Meanshift clustering algorithm is a practical application of the mode finding procedure: starting on the data points, run Meanshift procedure to find the stationary points of the density function, and prune these points by retaining only the local maxima. The set of all locations that converge to the same mode defines the basin of attraction of that mode. The points which are in the same basin of attraction is associated with the same cluster.

3) Random centers selection: When the amount of features extracted from videos is big enough, e.g. 100.000 or more, it is reasonable to select the centers of each cluster randomly. Iterating the selection after classification stage, the best result is kept, as it is done in RANSAC [13]. Due to its randomness, non-representative selection can occur. The iteration proceeding allows to avoid this bad sampling.

C. Bag of Visual Words and Classification

We base our approach to the traditional Bag of Visual Words, building a codebook and representing videos with words of this dictionary. In this work we limit the computation complexity using a subset of 100.000 uniformly selected samples to construct the codebook. We use a non-linear SVM classifier to recognize actions. We use a χ^2 kernel as in [2]

$$K(v_i, v_j) = exp(D(v_i, v_j)) \qquad (6)$$

where $D(v_i, v_j)$ is the χ^2 distance between video v_i and v_j. Since our action recognition is the multi-class classification, we use a one-against-all approach and determine the class with the highest confidence score.

III. Experimental Results

We performed the experiments with a 8x i7-2600 CPU at 3.40GHz. In section 3.1 we present a brief description of the dataset used for validating our method. In section 3.2 we evaluate the influence of the clustering parameters, and compare the results obtained for each proposed clustering method, using as a metric the final recognition performance.

A. Datasets

The KTH dataset [3] contains six different human actions: boxing, hand-waving, hand-clapping, walking, jogging and running. Each action is performed by 25 subjects in 4 different scenarios. In our experimental setup, we use three of the six

Fig. 2. KTH Dataset: boxing, hand-waving and running are used in our experiments.

actions (boxing, hand-waving and running) performed by randomly selected performers. We evaluate a SVM classifier for each of these actions using a one-against-all cross-validation. In Fig.2 we show the three actions taken from the dataset.

B. Codebook Size Influence

In order to see the influence of the codebook size, we compute nine different dictionaries sizes for K-means and random, of 100, 500, 1000, 1500, 2000, 2500, 3000, 3500 and 4000 words. Meanshift is used once, with a bandwidth of 3. This is due to we want a significant number of clusters, i.e. 4000. With this bandwidth value we get 3319 clusters.

As we can see in Fig.3, we need a large amount of words to get a good recognition. This means that clustering process is so crucial, and it usually takes a substantial quantity of time to compute this dictionary.

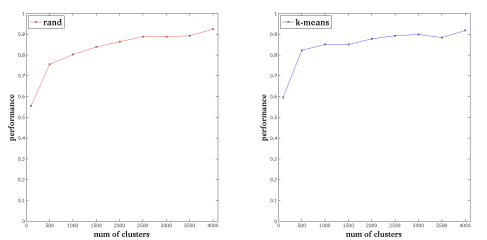

Fig. 3. Codebook size influence. The more number of words having dictionary the better action recognition.

C. Comparison between Methods

In our experiments, we follow the setup of [1], in which the HOG3D parameters are optimized for KTH Dataset. We have extracted Harris corners as IP due to these kind of points have shown better performance compared to DENSE sampling or FAST IP, as it can be seen in Table I. Using a K-means codebook of 4000 words, DENSE sampling with a 9x9 grid has given a 82.04% mean performance of action recognition, better than FAST points extraction, with a 81.66%. The best results are obtained with Harris IP, with a 91.50% of good recognition.

TABLE I
INTEREST POINTS EXTRACTION.

Method	Action 0	Boxing	Handwaving	Running
DENSE 9x9	83.3%	83.0%	79.4%	83.7%
Harris	93.0%	94.1%	92.6%	87.8%
FAST	87.3%	81.9%	82.1%	80.9%

For codebook generation, we have set the number of words to 1000 and 4000, for K-means and random selection, and we have tuned the Meanshift bandwidth accordingly to the number of clusters desired, i.e., around 4000 words. This bandwidth value has been set to 3. For K-means we guarantee the minimum clustering error by setting the iterations to 10, and for random selection we iterate also 10 times to guarantee the best random sampling.

As it is shown in Table II, Meanshift clustering takes the longest time to finally give similar results to others. It can be seen also that despite the variation in the number of words, K-means always outperforms random selection.

TABLE II
COMPARISON BETWEEN CODEBOOK GENERATION METHODS: K-MEANS, MEANSHIFT AND RANDOM SELECTION.

Cluster	$n_\circ$ clusters	$n_\circ$ iterations	performance	computation time (s)
K-means	1000	10	83.4%	689
K-means	4000	10	91.5%	4656
Meanshift	3319	-	89.8%	42156
random	1000	10	79.4%	0
random	4000	10	90.3%	0

IV. CONCLUSION

In this paper we analysed and compared three different methods to compute the code-

book of the traditional BoW approach. We used the final action recognition performance as the evaluation metric to carry out our purpose and, finally, we evaluated our framework on a public action dataset. We discussed the importance of the clustering parameters selection to determine their influence over the recognition results. In the end, we finally proposed to take into account the random selection, by which surprisingly good recognition performance is achieved. In the future work, we will build a semantic relationship between objects and actions, and we will need an efficient clustering that was fast and good enough for real-time applications.

ACKNOWLEDGMENT

This research has been partially supported by the Industrial Doctorate program of the Government of Catalonia, and by the European Community through the FP7 framework program by funding the Vinbot project (N°605630) conducted by Ateknea Solutions Catalonia.

REFERENCES

[1] A. Kläser and M. Marszalek, *A Spatio-temporal Descriptor Based on 3D-gradients.*, British Machine Vision Conference, 2008.

[2] I. Laptev and M. Marszalek, *Learning Realistic Human Actions from Movies.*, IEEE Conference on Computer Vision and Pattern Recognition, pages 1-8, 2008.

[3] C. Schuldt and I. Laptev and B. Caputo, *Recognizing Human Actions: a Local SVM Approach.*, International Conference on Pattern Recognition, pages 32-36, 2004.

[4] S. Lazebnik and 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, pages 21692178, 2006.

[5] Wang (Results) H. Wang and A. Kläser, *Action Recognition by Dense Trajectories.*, IEEE International Conference on Computer Vision and Pattern Recognition, pages 3169-3176, 2011.

[6] S. Wong and R. Cipolla, *Extracting Spatiotemporal Interest Points Using Global Information.*, IEEE 11th International Conference on Computer Vision, pages 18, 2007.

[7] X. Zhang and H. Zhang and X. Cao, *Action Recognition Based on Spatial-temporal Pyramid Sparse Coding.*, International Conference on Pattern Recognition, 2012.

[8] I.S. Dhillon and Y. Guan and B. Kulis, *Weighted Graph Cuts Without Eigenvectors a Multilevel Approach.*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 29, NO.11, pages 194457, 2007.

[9] M. Ester and H.P. Kriegel and J. Sander and X. Xu, *A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.*, Proceedings of 2nd International Conference on Knowledge Discovering and Data Mining, 1996.

[10] D. Comaniciu and P. Meer, *Mean Shift: A Robust Approach Toward Feature Space Analysis.*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 24, NO.5, pages 603619, 2002.

[11] J. MacQueen, *Some Methods for Classification and Analysis of Multivariate Observations*, Proc. Fifth Berkeley Symp. Math. Statistics and Probability, pages 281-296, 1967.

[12] P. Das and C. Xu and RF. Doell and JJ. Corso, *A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching*, IEEE International Conference on Computer Vision and Pattern Recognition, 2013.

[13] M.A. Fischler and R.C. Bolles, *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*, Comm. of the ACM 24 (6): pages 381395, 1981.

[14] G. Csurka and C. Dance and L. Fan and J. Willamowski and C. Bray, *Visual categorization with bags of keypoints*, ECCV workshop on Statistical Learning in Computer Vision, pages 59-74, 2004.