Bag-of-Words for Action Recognition using Random Projections
An Exploratory Study

Pau Agustí[1,3], V. Javier Traver[1,3], Filiberto Pla[1,3] and Raúl Montoliu[2,3]
[1]*DLSI, Jaume-I University, Castellón, Spain*
[2]*DICC, Jaume-I University, Castellón, Spain*
[3]*iNIT, Jaume-I University, Castellón, Spain*

Keywords: Clustering, Human Action Recognition, Random Projections, Bag-of-Words.

Abstract: During the last years, the bag-of-words (BoW) approach has become quite popular for representing actions from video sequences. While the BoW is conceptually very simple and practically effective, it suffers from some drawbacks. In particular, the quantization procedure behind the BoW usually relies on a computationally heavy *k*-means clustering. In this work we explore whether alternative approaches as simple as random projections, which are data agnostic, can represent a practical alternative. Results reveal that this randomized quantization offers an interesting computational-accuracy trade-off, because although recognition performance is not yet as high as with *k*-means, it is still competitive with an speed-up higher than one order of magnitude.

1 INTRODUCTION

Nowadays, human action recognition in videos has become an important research topic within the computer vision (Aggarwal and Ryoo, 2011), having a high impact on large technical and social applications. The goal in this field is the recognition from video sources of actions performed by individuals.

Some of the recent works are devoted to achieve view-invariance (Li and Zickler, 2012), learn from few examples (Natarajan et al., 2010), or explore higher-level action representations (Sadanand and Corso, 2012). Since, background segmentation is unrealistic, spatio-temporal interest points and its descriptors (Laptev, 2003) are one of the most successful approaches. However, these techniques tend to detect and describe too many non-discriminative points (i.e. points from the background, illumination changes, etc). To obtain more discriminative interest points, some approaches filter the interest points (Chakraborty et al., 2012).

As a complementary method to the interest points detection and description, the *Bag-of-Words* (BoW) approach has reached an important success in generating representation of those orderless features (Bilinski and Bremond, 2011; Wang et al., 2009), or including temporal order (Bregonzio et al., 2012).

Despite the advances and different approaches in the BoW schemes, *k*-means (Jain, 2010) is the most used clustering algorithm to construct vocabularies in these problems. Despite the fact that *k*-means is considered one of the most important data mining algorithm (Wu et al., 2007), the computation time is one of its drawbacks. Many works attempt to speed up the clustering processes. In (Boutsidis et al., 2010) random projections are used as a dimensionality reduction technique that allows *k*-means to have better computation time.

We interpret that in human action recognition, the role of clustering in BoW is not necessarily "semantic" (as in, e.g. image segmentation), but just for quantization and eventually, classification purposes. Given that randomized algorithms have interesting properties, the action representation could be based on them. Thus, this work aims at exploring random projections-based quantization as an alternative to *k*-means-based to generate vocabularies in a BoW scheme for action recognition.

Other related works use the concept of randomness to build vocabularies in a BoW model for recognition tasks, like (Moosmann et al., 2008) for images or (Mu et al., 2010) for human action recognition.

The approach presented here aims at using the concept of randomness to perform a clustering of feature points, keeping the initial idea of a *k*-means clustering to build a vocabulary, that is, to use clustering as way to quantize the original data set, but taking the advantage of the use of random projections to perform

Agustí P., Traver V., Pla F. and Montoliu R..
Bag-of-Words for Action Recognition using Random Projections - An Exploratory Study.
DOI: 10.5220/0004216906140619
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 614-619
ISBN: 978-989-8565-47-1

the clustering in a more efficient way as k-means does. Some preliminary work is presented in this paper to analyze the performance both in terms of computational time and recognition rate in comparison with k-means, widespread used and the *de facto* standard for quantization in BoW models.

2 METHODOLOGY

The methodology used follows the classic BoW STIP-based (Spatio Temporal Interst Point) procedure. An interest point detector is applied to each video (using Harris3D from (Laptev, 2003)) in order to obtain a descriptor for each point of interest. The descriptors extracted are, also from (Laptev, 2003), the Histograms of Gradients (HOG), the Histograms of Optical Flow (HOF) and a concatenation of both (HOGHOF). Once the descriptors are extracted, a histogram is generated for each video. This part will be tested using two different approaches to get the quantization of the histogram. The traditional k-means-based quantization, where descriptors are grouped into clusters and the histogram is a quantization of how many times each descriptor, from a video, belongs to a cluster. The second approach is the random projection-based quantization, where the histogram is done using random matrices. At the end of the process, the whole database is split into training set, validation set and test set. Finally, an SVM, with a χ^2 kernel, is trained. The recognition rate is obtained by using the test set.

2.1 k-means-based Quantization

In the BoW scheme, the most common way to create the histogram quantization is the use of k-means clustering (Bilinski and Bremond, 2011; Wang et al., 2009). In this approach, we will refer to the procedure having two steps. First, all input data ($\mathbf{SS}_{M\times d}$) is grouped into a number (k) of clusters and the centroids ($\mathbf{CC}_{k\times d}$) of them are given as output (Algorithm 1). The asymptotic cost of this step is $O(\gamma Mdk)$, where γ is the number of iterations to reach the convergence, d is the dimensionality of original space and M the number of examples to be clustered. The second step (detailed in Algorithm 2), starts with the data from a video (where N is the number of features describing the video), and create a histogram of k bins. For each feature vector, the closest centroid is found and this contributes to the corresponding bin number. The asymptotic cost of this step is $O(Ndk)$.

Despite the great acceptance that the technique has in the human action recognition field, the computation time to perform the clustering is very costly. Indeed, it is shown that the problem to optimize the cost function in the classical k-means algorithm is a NP-hard problem (Aloise et al., 2009).

Algorithm 1: Finding the clusters by k-means (Step 1).

Input: Dimensionality of original space, d,
number of examples, M,
number of clusters, k, and
Input data matrix, $\mathbf{SS}_{M\times d}$
Output: Clusters centroids $\mathbf{CC}_{k\times d}$,
1: Randomly initialize the cluster centroids $\mathbf{CC}$
2: **repeat**
3: **for** $i \leftarrow 1$ to M **do**
4: Assign $\mathbf{SS}_i$ to the closest $\mathbf{CC}$
5: **end for**
6: **for** $i \leftarrow 1$ to k **do**
7: $\mathbf{CC}_i \leftarrow$ centroid of the points assigned to the i-th cluster
8: **end for**
9: **until** convergence

Algorithm 2: Generating the histograms (Step 2).

Input: Dimensionality of original space, d,
number of clusters, k,
number of examples, N,
Centroids of clusters, $\mathbf{CC}_{k\times d}$, and
Input data matrix, $\mathbf{X}_{N\times d}$
Output: The histogram $\mathbf{h}_{1\times k}$,
1: **for** $i \leftarrow 1$ to N **do**
2: Let j be the closest cluster centroid of data in i-th row of $\mathbf{X}$
3: $\mathbf{h}_j \leftarrow \mathbf{h}_j + 1$
4: **end for**
5: $\mathbf{h} \leftarrow \frac{\mathbf{h}}{N}$

2.2 Random Projections-based Quantization

Random projections have been reported (Bingham and Mannila, 2001) to be a competitive alternative to other dimensionality reduction techniques such as PCA, yet computationally simpler. The rationale behind random projections is the Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss, 1984) which states that distances between points in a given space are preserved after they are randomly projected to a space of suitable high dimensionality (see (Bingham and Mannila, 2001) and references within it). Clustering of data after their random projections has been used in the past. For instance, the use of EM clustering is particularly suitable in conjunction with random projections because EM assumes data are distributed

as mixture of Gaussians, and high-dimensional data are more Gaussian after being randomly projection to low-dimensionality space (Fern and Brodley, 2003).

In the context of our problem, we are not interested in dimensionality reduction nor in clustering *per se*, but in obtaining a histogram-like bag-of-words representation. The proposed procedure to compute these histograms using random projections has two steps. First, a random projection matrix $\mathbf{R}$ and a partition matrix $\mathbf{H}$ are computed (Algorithm 3). Importantly, this step is independent of the data to be clustered and only its dimensionality d is required, along with three parameters (P, L, and b). Parameter P is the number of projections and can also be viewed as the dimensionality of the projected space. Parameter L is the number of partitions that are considered. Each partition is a combination of b out of the P projections. The histograms will have 2^b bins. The entries of the projection matrix are generated from the standard normal distribution $\mathcal{N}(0,1)$. The asymptotic cost of this step is $O(Pd+Lb)$.

Algorithm 3: Generating the projection and partitioning matrices (Step 1).

Input: Dimensionality of original space, d,
number of projections, P,
number of partitions, L, and
number of bins of target histogram, 2^b
Output: the projection matrix $\mathbf{R}_{P\times d}$, and
the partition matrix $\mathbf{H}_{L\times b}$
1: Fill in $\mathbf{R}$ with random numbers from $\mathcal{N}(0,1)$
2: Fill in each row of $\mathbf{H}$ with b random integers in $\{1,P\}$ without any repetition.

The second step (detailed in Algorithm 4) uses a data matrix $\mathbf{X}$, which is projected and (implicitly) clustered. Each data vector in $\mathbf{X}$ is projected using the previously computed projection matrix $\mathbf{R}$, and contributes to an histogram entry using the partition matrix $\mathbf{H}$. For N data points in $\mathbf{X}$, this step is $O(N(Pd+Lb))$.

For STIP-based action recognition, d is the dimensionality of the descriptor of the interest points, and matrices $\mathbf{R}$ and $\mathbf{H}$ can be generated for some values of P, L, and b. Data $\mathbf{X}$ would correspond to all the descriptors computed on an action sequence. For given matrices $\mathbf{R}$ and $\mathbf{H}$, it is expected that the histograms resulting from sequences of the same action will look more similar than those of sequences of different actions.

2.3 Complexity Comparison

The computation complexity has been split into two steps, each step corresponds to each one of the two

Algorithm 4: Projecting and quantizing the input data (Step 2).

Input: Input data matrix, $\mathbf{X}_{N\times d}$,
the projection matrix $\mathbf{R}_{P\times d}$, and
the partition matrix $\mathbf{H}_{L\times b}$
Output: the histogram $\mathbf{h}_{1\times 2^b}$, and
1: $\mathbf{Z} \leftarrow (\mathbf{R}\mathbf{X}^T) > 0$ {Project data and binarize}
2: **for** $i \leftarrow 1$ to N **do**
3: $\mathbf{z} \leftarrow i$-th column of $\mathbf{Z}$
4: **for** $j \leftarrow 1$ to L **do**
5: $l \leftarrow \sum_{k=1}^{b} 2^k \cdot \mathbf{z}_{\mathbf{H}_{jk}}$ {Get bin index}
6: $\mathbf{h}_l \leftarrow \mathbf{h}_l + 1$ {increase histogram count}
7: **end for**
8: **end for**
9: $\mathbf{h} \leftarrow \frac{1}{LN}\mathbf{h}$ {Normalize histogram}

algorithm explained in the respective sections. If we assume that $Pd \approx Lb$, the second step in the random projection-based quantization does not give any advantage, because in the worst case both are cubic order and depends on the amount of data. The really great advantage for the random projections-based quantization is in the first step. The cost of k-means clustering is, in the worst case, a polynomial of degree 4 and the random projections-based is only a polynomial of degree 2. Another drawback, in this step, is the k-means-based dependence of the amount of data, N, what makes this method inconvenient when dealing with high amount of data.

3 EXPERIMENTAL WORK

For the experiments, the Weizmann and KTH datasets have been used. These datasets are ones of the most used for human action recognition. They are publicly available, what allows comparison with different approaches. The Weizmann dataset (Gorelick et al., 2007) contains 93 sequences showing 9 different people performing 10 different actions. The KTH dataset (Schüldt et al., 2004) contains 599 sequences showing 25 people performing 6 actions in 4 different scenarios.

In this work, to study the feasibility of random projections for quantization applied to human action recognition, these following issues will be tested:

- Comparison of the runtime between k-means-based quantization and random projections-based quantization to generate the histograms (Section 3.1).
- Are the recognition results, using a random projection-based quantization, comparable to the

state-of-art using a k-means-based quantization? (Section 3.2)

- As the method is random by nature, how does this affect the stability of the results? (Section 3.3)
- How does the parameters configuration influence to the random projection-based quantization? (Section 3.4)

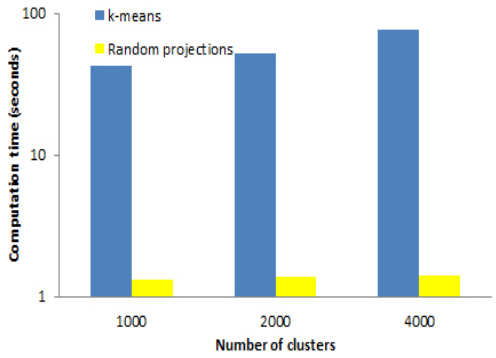
For evaluating the recognition performance, the learning-classification protocol follows the same conditions as in (Bilinski and Bremond, 2011), which is chosen as a reference for k-means results. For the Weizmann dataset, a *leaving-one-actor-out* protocol is used and for the KTH dataset, persons 2,3, 5–10 and 22 are used for the test set and the rest for the training set.

3.1 Computation Time

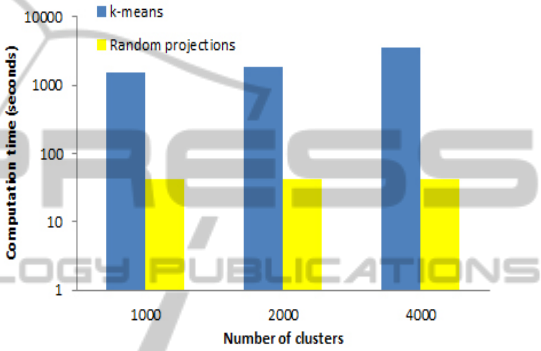
As shown in Section 2.3, theoretical the computational complexity is lower when using a random projection-based quantization than in a k-means-based quantization. Nevertheless, does the *running time* decrease? To answer this question, as the dimensionality (d) has effect on the clustering, the objective was to penalize as much as possible the computational time, so a concatenation of the HOG and HOF descriptor has been chosen (HOGHOF).

Figure 1 shows a comparison of the runtime using the k-means-based quantization and that of the random projections-based quantization for the Weizmann dataset (a) and the KTH dataset (b). The runtime is calculated for the two steps together: (1) definition of the clusters (k-means-based) or generation of the random matrices (random projections-based) and (2) generation of the histograms. For k, the values 1000, 2000 and 4000 clusters has been tested and the closest values for the b in random projections-based are 10, 11 and 12 (1024, 2048 and 4096 bins), and fixed values for $L = 200$ and $P = 250$.

For the Weizmann dataset, the time elapsed with random projection is always less than 2 seconds and using the k-means is always more than 40 seconds. As it has been shown, k-means is really dependent on the number of data, M. Indeed, following some works like (Wang et al., 2009) to reduce the complexity, the clustering has been performed using only a part of the data (in that case, and in this work, $\approx 100,000$ descriptors) for the KTH dataset. The runtime to create the histograms using k-means is always over 25 minutes and using the random projections-based quantization is always less than 42 seconds. Nevertheless, the times to generate the histograms only (the second step, it has been shown also in Section 2.3) are almost the same.



(a) Weizmann dataset



(b) KTH dataset

Figure 1: Computation time in logarithmic scale for different number of clusters.

The speed-up factor of using random projections over using k-means is about 30–50 in Weizmann, and about 35–80 in KTH. Thus, the proposed clustering is more than one order (and can be up to almost two orders) of magnitude faster than k-means.

3.2 Accuracy

Once it is known that using a random projections-based quantization instead a k-means, the computation time is a significant advantage, the recognition rate using random projection should be assessed with respect to the k-means-based algorithm. Thus, for the experiments, the k-means based BoW algorithm proposed in (Bilinski and Bremond, 2011) is used.

Table 1 shows the best results obtained by (Bilinski and Bremond, 2011) for different descriptors using a k-means-based quantization (they test with 1000, 2000, 3000 and 4000 clusters) and the best results obtained with our method (the parameters configurations tested for choosing the best will be reported in section 3.4). The accuracy obtained just by changing the histogram generation is, in most of the cases, only around 3% less. And in comparison to (Kläser et al., 2008) for Weizmann and (Wang et al.,

2009) for KTH (the same strategy and k-means-based quantization ara used) our recognition rate is always (for all combination of HOG/HOF descriptors) higher except for the HOG descriptor in KTH.

3.3 Stability

Assuming the random nature due to the generation of random matrices, the question about the stability of the results should be addressed. To perform this experiment, the HOF descriptor has been chosen due to the good results obtained in the experiments in Section 3.2. For the same parameters configuration (b, L, P), 20 different random matrices were generated and used for the random projection-based quantization classification. In both datasets, the best result and the worst were used, according to the experiments in Section 3.2.

Table 2 shows the mean of the recognition rates and its standard deviation. It could be appreciated there is not a significant accuracy variation using the same parameters configuration. It is worth mentioning that the Weizmann database approximately has 10% of difference between the best and the worst result. In the case of the KTH dataset this difference is more than 75%, what introduces the question about the importance of the selection of parameters.

3.4 Configuration Dependency

In order to assess the dependency of the random projections on the configuration of the parameters, in the experiments the following values have been used $b \in \{8,9,10,11\}$, $L \in \{5,10,25,50,100,150,200\}$ and $P \in \{50,100,150,200,250\}$ and all the combinations among them. Regarding the accuracy, experiments show that only few parameters configurations provided the best results, thus, the selection of parameters is really important.

Figure 2 shows in the left side the Weizmann (HOF descriptor) results varying only one parameter in each plot (the other two were fixed at $b = 8$, $L = 200$ and $P = 250$ depending on which one is varying). In the right side it is shown the KTH (HOF descriptor) results varying only one parameter in each plot (the other two were fixed at $b = 12$, $L = 100$ and $P = 50$ depending on which one is varying).

Despite the fact we can notice some pattern in some cases (in Weizmann accuracy increase while increasing the L and P parameters and in KTH the accuracy increase while increasing b), it seems there is not a clear behavior rule. Therefore, in order to choose the parameters configuration, it seems necessary to select them by a validation method.

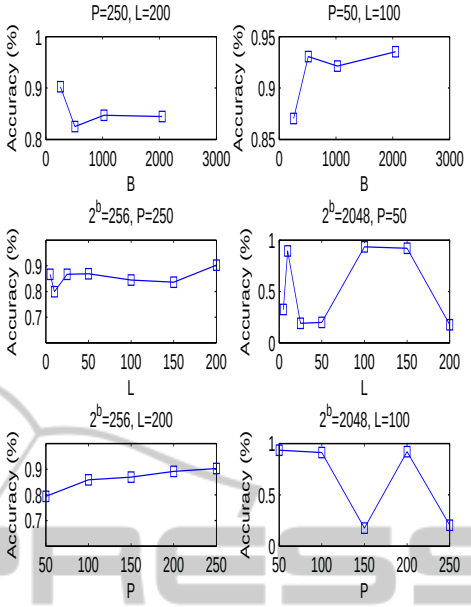


Figure 2: Influence of different parameters configurations on the recognition accuracy (Left for Weizmann, right for KTH).

4 CONCLUSIONS

This papers has presented the use of random projections-based quantization as an alternative to k-means-based clustering for building vocabularies using a bag-of-words representation for action recognition. A preliminary work to assess the performance of the method with respect to the widespread used k-means-based algorithm has been done.

One clear advantage of the proposed random projections-based algorithm is that it is computationally more advantageous than the k-means. The reason for this efficiency is that, unlike k-means, the random projections-based algorithm does not make any use of data to define the clustering. Regarding the action recognition performance, competitive rates are achieved, but those with k-means-based clustering are generally better. Therefore, the proposed quantization based on random-projection represents an interesting trade-off between computational effort and accuracy. Further work is directed to boost this approach with quantization mechanisms that offer both high recognition performance and great computational benefit.

ACKNOWLEDGEMENTS

This work is partially supported by the Spanish research programme Consolider Ingenio-2010

Table 1: Recognition results (%) with the two BoW schemes explored.

(a) Weizmann dataset

	k-means		Random Projections				
Descriptor	*k*	**Accuracy**	*b*	*L*	*P*	**Accuracy**	**Difference**
HOG	2,000	86.02	512	100	200	82.47	3.55
HOF	3,000	91.40	256	200	250	90.25	1.15
HOGHOF	2,000	92.47	2,048	100	250	90.00	2.47

(b) KTH dataset

	k-means		Random Projections				
Descriptor	*k*	**Accuracy**	*b*	*L*	*P*	**Accuracy**	**Difference**
HOG	1,000	83.33	1,024	200	250	70.00	13.33
HOF	1,000	95.37	2,048	100	50	93.51	1.86
HOGHOF	3,000	94.44	2,048	200	200	92.12	2.32

Table 2: Stability results obtained by repeating 20 times each experiment with the same parameters configuration (b, L, P).

	Best result					**Worst result**				
	Mean (μ)	Std. dev. (σ)	*b*	*L*	*P*	Mean (μ)	Std. dev. (σ)	*b*	*L*	*P*
Weizmann	88.50	1.78	8	200	250	80.33	2.22	11	5	50
KTH	91.79	1.73	11	100	50	14.76	4.03	11	25	250

CSD2007-00018, Fundaci Caixa-Castell Bancaixa (P11A2010-11 and P11B2010-27) and Generalitat Valenciana (PROMETEO/2010/028).

REFERENCES

Aggarwal, J. and Ryoo, M. (2011). Human activity analysis: A review. *ACM CS*.

Aloise, D., Deshpande, A., Hansen, P., and Popat, P. (2009). NP-hardness of euclidean sum-of-squares clustering. *ML*.

Bilinski, P. and Bremond, F. (2011). Evaluation of local descriptors for action recognition in videos. In *ICCVS*.

Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In *KDD*.

Boutsidis, C., Zouzias, A., and Drineas, P. (2010). Random Projections for k-means Clustering. *NIPS*.

Bregonzio, M., Xiang, T., and Gong, S. (2012). Fusing appearance and distribution information of interest points for action recognition. *PM*.

Chakraborty, B., Holte, M. B., Moeslund, T. B., and Gonzlez, J. (2012). Selective spatio-temporal interest points. *CVIU*.

Fern, X. Z. and Brodley, C. E. (2003). Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach. In *ICML*.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as Space-Time Shapes. *tPAMI*.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *PRL*.

Johnson, W. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. In *CMAP*.

Kläser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *BMVC*.

Laptev, I. (2003). On space-time interest points. *IJCV*.

Li, R. and Zickler, T. (2012). Discriminative virtual views for cross-view action recognition. In *CVPR*.

Moosmann, F., Nowak, E., and Jurie, F. (2008). Randomized Clustering Forests for Image Classification. *tPAMI*.

Mu, Y., Sun, J., Han, T. X., Cheong, L.-F., and Yan, S. (2010). Randomized locality sensitive vocabularies for bag-of-features model. In *ECCV*.

Natarajan, P., Singh, V. K., and Nevatia, R. (2010). Learning 3D action models from a few 2D videos for view invariant action recognition. In *CVPR*.

Sadanand, S. and Corso, J. J. (2012). Action bank: A high-level representation of activity in video. In *CVPR*.

Schüldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local SVM approach. In *ICPR*.

Wang, H., Ullah, M. M., Kläser, A., Laptev, I., and Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *BMVC*.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., and Steinberg, D. (2007). Top 10 algorithms in data mining. *KAIS*.