Motion Words for Videos

Ekaterina H. Taralova, Fernando De la Torre, and Martial Hebert

Carnegie Mellon University, USA

Abstract. In the task of activity recognition in videos, computing the video representation often involves pooling feature vectors over spatially local neighborhoods. The pooling is done over the entire video, over coarse spatio-temporal pyramids, or over pre-determined rigid cuboids. Similarly to pooling image features over superpixels in images, it is natural to consider pooling spatio-temporal features over video segments, e.g., supervoxels. However, since the number of segments is variable, this produces a video representation of variable size. We propose Motion Words - a new, fixed size video representation, where we pool features over supervoxels. To segment the video into supervoxels, we explore two recent video segmentation algorithms. The proposed representation enables localization of common regions across videos in both space and time. Importantly, since the video segments are meaningful regions, we can interpret the proposed features and obtain a better understanding of *why* two videos are similar. Evaluation on classification and retrieval tasks on two datasets further shows that Motion Words achieves state-of-the-art performance.

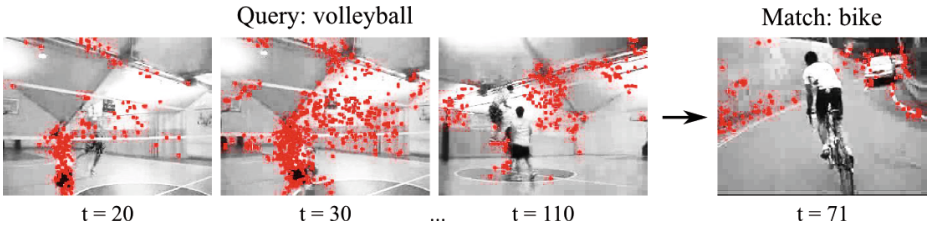
Keywords: Video representations, action classification.

1 Introduction

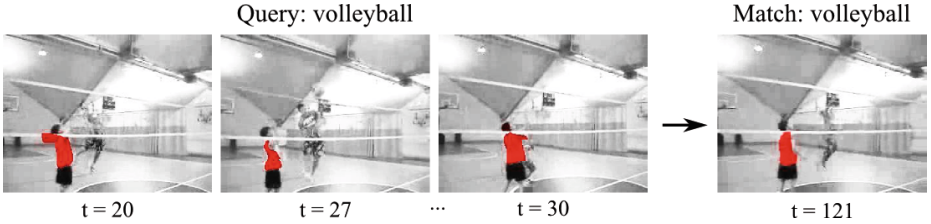
Features for video classification and retrieval include low-level interest point features [33,34,8,4], mid-level patch-based features [15,1,35,38], and higher level, semantic features [22]. Even though low-level features are limited either in temporal scale [33,8], or in density [4], they robustly capture local information, and in fact obtain state-of-the-art classification performance on several datasets [23,17]. However, if we want to know why two videos are classified as similar, visualizing the low-level features does not allow us to interpret the results. On the other hand, mid-level video patches also perform well for activity classification [15,6,35], and we can visualize the cuboids learned as important for classification. These representations have been limited to cuboids of predetermined spatial and temporal sizes [15], or rectangles from object/foreground detectors [40]. High-level features provide semantic interpretation at the expense of additional annotations or training [22].

Ultimately, we seek a video representation that captures both low-level and region-based statistics. Furthermore, it is important that the representation enables interpretability. That is, when visualized, we want features that give us the power to understand which regions make two videos similar. We propose a video

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 725–740, 2014.



(a) In BoW, features from one codebook center originate from different regions



(b) In the proposed BMW, supervoxels from one Motion Words codebook center originate from similar regions, and are easy to interpret

Fig. 1. In standard BoW pooling (a), features match from regions of very different appearance and motion. In the proposed BMW representation (b), features pooled over supervoxels match similar regions and enable interpretability.

representation in which we pool low-level features over regions defined by a video segmentation. It is a natural idea to pool features over coherent spatio-temporal regions, e.g., supervoxels. In fact, work in image analysis shows that descriptors computed over segmentation regions (e.g., superpixels) provide more robust image representations [2]. Nevertheless, in video analysis, pooling is currently done either over the entire video [33,8], over coarse spatio-temporal pyramids [33], or over pre-determined rigid cuboids [19,9,27]. Each video is then represented by the concatenation of the regions.

However, pooling is not necessarily local in the feature vector space and widely dissimilar features may be pooled together [2]. We visualize such a scenario in Figure 1, where volleyball players and their interactions with the ball occur at various spatio-temporal locations in a video from the Youtube [23] action dataset. We take this video as a query and ask for the nearest neighbor from the dataset using the standard Bag of Words framework with state-of-the-art Dense Trajectory [33] features. The dataset contains a very similar video of the same players, in the same environment, performing a different golf swing trial. We are thus puzzled when the system retrieves an incorrect result, an outdoors "biking" video. If we pick a BoW codebook center and visualize the features that are encoded by this center in both the query and the match, we see that the features come from video regions with very different motion and appearance, e.g., players, wall, sidewalk, car (Figure 1a). While we can peek into BoW in this manner, this visualization does not provide any intuition as to what makes the videos similar.

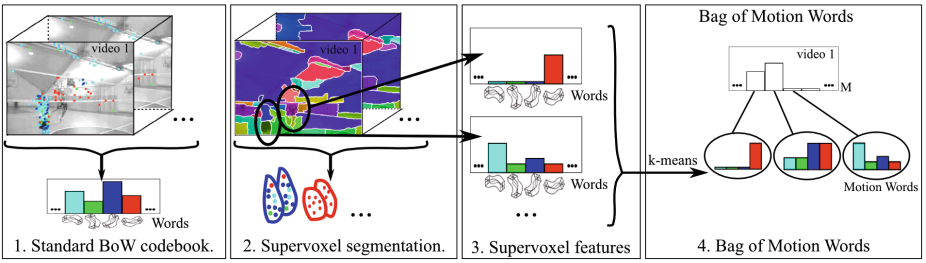


Fig. 2. In the proposed Bag of Motion Words framework we start with a standard BoW codebook (e.g., computed from Dense Trajectories [33]), compute a supervoxel segmentation, and pool the encoded low-level features over the supervoxels. We cluster the supervoxel-based feature vectors using k-means to learn a codebook of Motion Words.

On the other hand, video segments provide more flexible spatio-temporal support than cuboids of manually chosen spatial and temporal sizes. For example, in the above scenario, when we pool features over supervoxels, we obtain a much better match - the expected "volleyball" video (Figure 1b). However, each video can have a different number of segments, resulting in video representations of variable sizes. In this paper, we propose a simple way of constructing a fixed-size representation by using the popular Bag of Words (BoW) framework. Rather than constrain all videos to have the same number of regions, we treat each video segment as a feature vector and cluster the segments from training videos to learn Motion Words. Each video is then represented as a Bag of Motion Words (BMW), as shown in Figure 2. Furthermore, the proposed Motion Words representation enables localization and interpretation. Since segmentation algorithms produce meaningful spatio-temporal regions, we can visualize and interpret the "words" that are common to both videos (Figure 1, bottom).

We present the method overview and its components in Section 3. In Section 4 we discuss design choices and experimental setup. We evaluate the representation qualitatively in Section 5, and quantitatively in Section 6.

2 Related Work

2.1 Feature Pooling

Pooling is one of the key steps in computing video representations. For example, when applied to videos, the Bag of Words representation is computed either by pooling features in an unstructured way over the entire video [33,8,28], over a coarse spatio-temporal pyramid [33,28], or over predetermined cuboids chosen for convenience or computational reasons [19,9,27]. Pooling low-level features over cuboids is also a key step to many methods that learn mid-level representations [15,6,22,35]. Le *et.al.* [19] automatically learn features from video data over predetermined cuboids, which are also used at pooling time. Recent works

use cuboids defined by users' gaze [25], or consider foreground cuboids, e.g., by detecting regions of interest [40]. To enable more robust spatio-temporal support, we propose to use an initial oversegmentation into coherent spatio-temporal regions, which is similar to using superpixel segmentation as the pre-processing step for image analysis [26].

The idea of using an initial over-segmentation has been explored in the image analysis community. For example, Gould *et.al.* [12] use superpixels as the basic data layer for decomposing a scene into geometric and semantically consistent regions. Similarly, Tighe [31] propose nonparametric image parsing with superpixels. Other works restrict pooling to inputs close in input space [16,39]. Image processing and de-noising works consider similar inputs to smooth noisy data over a homogeneous sample without throwing out the signal [5,7,24]. We hypothesize that in videos it is important to pool features locally in space and time, e.g., over supervoxels, which are regions coherent in motion and appearance.

In video analysis, works that first compute supervoxels followed by various task-specific processes include hierarchical grouping [13], long-range tracking [3,21], superpixel flow [32] and mid-level features [9]. On the other hand, Zhang *et.al.* [38] model combinations or co-occurances of low-level features, Essa *et.al.* [1] use n-grams and regular expressions to encode long-term motion information. Zhang *et.al.* [38] propose mid-level features that rely on the definition of a correspondence transform to compare videos with variable number of regions. Rather than develop new metrics to compare videos with a variable size representation, we simply perform a second clustering step to learn a codebook of the region-based features. This provides a fixed size representation for videos which can be used in standard classification methods.

2.2 Video Segmentation

In recent work on video segmentation, Brendel and Todorovic [3] segment videos into spatiotemporal tubes designed to represent moving objects. They propose a simple, blocky segmenter, which uses compression error to split and then merge image regions in a small temporal window, based on HSV color values and Lucas-Kanade optical flow. Others attempt to segment foreground objects while avoiding over-segmentation [20]. Grundmann [13] propose a hierarchical graph-based segmentation which extends the Felzenszwalb and Huttenlocher [10] method for segmenting images. They build a graph of color regions and connect them over time based on color and motion histogram distance. Xu *et.al.* [37] create a streaming version of this method that is computationally much more efficient and can be applied to videos with larger number of frames. The recent Uniform Entropy Slice [36] method provides a way to select supervoxels from different hierarchies of the segmentation based on a user-defined feature criterion, for example, "motionness." Selecting regions across the hierarchy alleviates the issue of under-segmentation at coarse levels and over-segmentation at fine levels.

To encode long-range motion cues, other algorithms build upon clusters of long trajectories [4]. Extending this work, Lezama *et.al.* [21] augment trajectories with local image information and seek a segmentation that respects

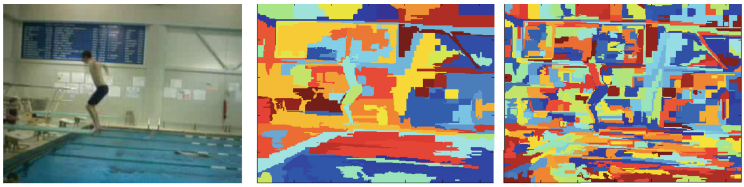


Fig. 3. Example supervoxels from the coarse and fine segmentation hierarchies of the streaming GBH algorithm [37]

object boundaries and associates these objects across frames. Raptis [29] also use clusters of long-term point trajectories, but require annotated bounding boxes and assume a fixed number of parts. Long-range trajectories are sparse and thus these methods ignore background motion, which is often very informative. Furthermore, finding a good track clustering function is essential, but not straightforward.

3 Motion Words

While pooling over rigid cuboids provides computational efficiency, it is natural to consider pooling features over more flexible spatio-temporal regions. Supervoxel segmentation algorithms provide excellent spatio-temporal support for feature pooling. Supervoxels are regions coherent in both appearance and motion over time, e.g., the streaming GBH segmentation [37] and the UES [36] methods (Figure 3). We propose a new video representation, Motion Words, where we pool low-level features over such coherent spatio-temporal regions. One way to represent a supervoxel would be to average the low-level descriptors within the supervoxel. However, averaging descriptors like HOG, HOF, MBH, STIPs, *etc.* with their neighbors results in the loss of a considerable amount of information [2]. Instead, we first encode the low-level descriptors to a standard Bag of Words codebook and average the codes within each supervoxel.

We construct the Motion Words representation in four steps (see Figure 2):

1. Compute a standard Bag of Words codebook from low-level descriptors;
2. Compute a supervoxel segmentation;
3. Pool the encoded low-level descriptors in each supervoxel to obtain supervoxel-based feature vectors.
4. Cluster the supervoxel-based vectors to learn a codebook of Motion Words.

3.1 Low-Level Features

Motion Words can be built from a variety of features and their combinations. For example, we can easily pool dense features (e.g., MBH [33], STIPs [18]) by simply counting those that fall within each supervoxel. On the other hand, features that

span several frames, e.g., Dense Trajectories [33,34], with default length of 15 frames, can be pooled by defining a minimum temporal overlap threshold with a supervoxel to determine which trajectories should be counted. Similarly, we can pool cuboid-based features (e.g., automatically learned features via subspace analysis, ISA [19]) by defining a minimum volume overlap threshold with a supervoxel. While we can also use long trajectories (e.g., Brox and Malik [4]), or features extracted only at interest points, they are sparse and many supervoxels will be empty. Nevertheless, such sparse features can be used to complement dense features. In the experimental section we report performance using Dense Trajectories, Dense Descriptors and ISA features.

3.2 Video Segmentation

Video segmentation algorithms provide an unsupervised way to generate coherent spatio-temporal regions, which, while not necessarily corresponding to objects, are easy to interpret. We seek a segmentation into supervoxels of sizes determined by appearance and motion cues, and not necessarily regions that respect object boundaries. That is, rather than impose a fixed number of regions or fix their size, we allow the method to find the best segmentation for each video. Video segmentation algorithms that optimize jointly for appearance and motion at the pixel level, e.g., [13,37], are excellent first choices to consider for generating supervoxels for Motion Words. One property that we hypothesize is essential in the context of Motion Words is a stronger emphasis on respecting motion boundaries as opposed to respecting appearance boundaries. The Unified Entropy Slice (UES) [36] segmentation method provides a way to do so by selecting supervoxels across the segmentation hierarchy levels that optimize a user-specified property. In the case of Motion Words, the "motion-ness" property is most relevant, where the method optimizes for motion boundaries based on optical flow. In the experimental section we use the freely available streaming GBH method [37] and compare Motion Words obtained using the most coarse level of segmentation, the finest level, the union of three hierarchy levels, and the UES [36] segmentation.

3.3 Bag of Motion Words (BMW)

Each video is an unordered collection of a variable number of supervoxel-based feature vectors of the same dimension (k). There are several ways we can proceed to construct the video representation. For example, Zhang *et.al.* [38] develop an approach to handle a variable number of features per video by defining a correspondence transform for comparing videos. Instead, we propose to use the statistical power of the Bag of Words framework a second time (Figure 2). We cluster the supervoxel features of the training videos to learn a Motion Words codebook. The video representation is a Bag of Motion Words (BMW), that is, a normalized histogram of Motion Word counts.

Formally, let κ be a codebook of size k learned over a set of features in a standard BoW framework (e.g., Dense Trajectories [33], or features learned

directly from video data [19]). Let $\Gamma^v = \{\gamma_1^v, \ldots, \gamma_n^v\}$ be a segmentation of video v into n spatio-temporal regions (e.g., obtained using the hierarchical method of Grundmann and Essa [13], or the streaming method of Xu *et.al.* [37]). We encode the low-level features to the codebook κ, and pool them within each region γ_i^v. That is, each supervoxel γ_i^v is represented as a histogram of size k (counts of the encoded low-level features within γ_i^v). We cluster the supervoxel histograms to learn a codebook of Motion Words of size M. Finally, the supervoxel histograms are encoded to the Motion Words codebook and each video v is represented as a histogram $\mathcal{M} = \{\mu_1, \ldots, \mu_M\}$ of Motion Word counts (see Figure 2).

4 Experimental Setup

There are several key design choices involved in building Motion Words. We discuss and evaluate the choice of low-level features, segmentation, quantization, and classifier methods.

4.1 Choice of Low-Level Features

For the underlying Bag of Words we consider three types of features that have been successfully used in activity classification: state-of-the-art Dense Trajectories (*DTs*) [33], automatically learned features through independent subspace analysis (*ISA*) [19], and dense HOG, HOF and MBH descriptors [33]. We extract features using code provided by the authors. Since we want to test pooling of different types of descriptors, we use the default settings without performing any parameter tuning.

4.2 Choice of Supervoxels

For the choice of spatio-temporal regions, we consider state-of-the-art video segmentation methods. The streaming graph-based algorithm of Xu *et.al.* [37] is well suited for Motion Words because it encodes properties such as spatio-temporal uniformity and coherence, and boundary detection. We find that the default parameters suggested by the authors are a good trade-off between size of supervoxels and motion boundary preservation. We extract three hierarchy levels and compare pooling over supervoxels from the coarsest level (GBH coarse), the finest level (GBH fine), and the union of all three levels (GBH combined).

Furthermore, we evaluate the Bag of Motion Words framework using the Uniform Entropy Slice segmentation algorithm [36], choosing to optimize for "motion-ness." Each video has $300-1000$ supervoxels generated from the streaming GBH method at the finest level, $20-100$ generated at the coarsest level, and $20-100$ supervoxels generated by the UES method. We randomly sample $100,000$ of the training supervoxels to learn a BMW codebook using k-means and Euclidean distance. We count trajectories as part of a supervoxel if at least half of the trajectory is contained in the supervoxel. In the case of ISA features, we count only those that overlap a supervoxel by at least 30%.

(a) Even when visualized, it is difficult to understand why DTs (shown in red) from the same codebook center originate from regions of different appearance and motion (e.g., static grass and mountains, and twisting torso). Best viewed in color.

(b) When visualized, Motion Words are easy to interpret - the SVs that quantize to the same codebook center as the manually chosen body region also correspond to the golfer's body. Best viewed in color.

Fig. 4. We manually select one point on the golfer's body and visualize all other descriptors that quantize to the same codebook center in two golf swing trials. Ideally, all descriptors will correspond to regions of the golfer's body.

Since extracting supervoxels is independent of the low-level descriptors, it can be done in parallel to the feature extraction. In our experiments, the time to segment videos took 1.3 times longer on average than extracting DTs.

4.3 Choice of Quantization Method

For computing the low-level feature codebook, we follow the setup of Wang *et.al.* [33] by randomly sampling 100,000 data points per feature channel, and clustering with k-means. To analyze the sensitivity of Motion Words we evaluate codebooks of sizes 5000, 1000, 500, and 200. Finally, for the Bag of Motion Words we consider k-means with 5000, 2500 and 1000 clusters using Euclidean distance.

Since the number of supervoxels per video can be very small, soft quantization is better suited in the encoding step. The smaller number of descriptors and their sparsity make the second quantization step much faster to compute than standard BoW. In our experiments, k-means for BMW took 0.3 the time to cluster DTs with the same number of codebook centers.

In our initial evaluation, we chose to pool the quantized supervoxel features over the entire video, not encoding temporal relationships across supervoxels. In

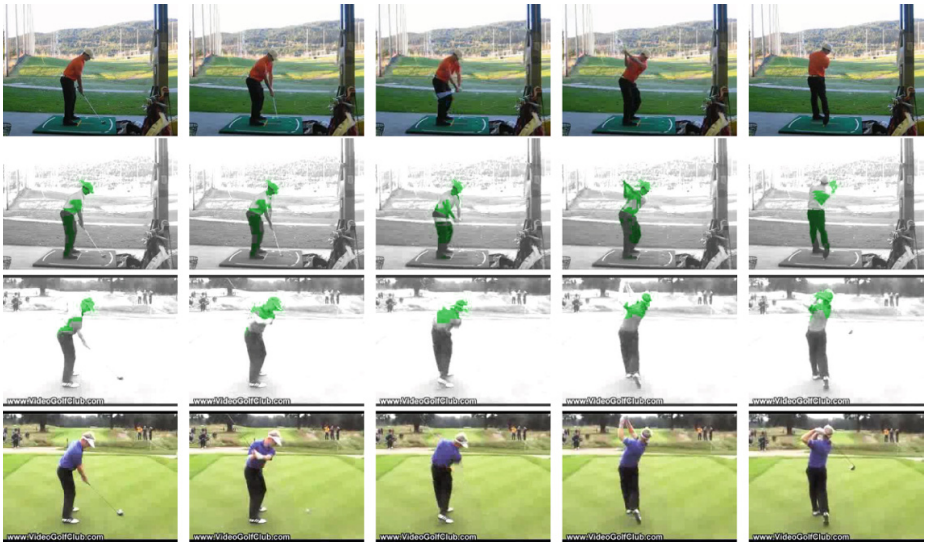


Fig. 5. Given a manually selected supervoxel on the golfer's body in the first video, we can visualize all other supervoxels that quantize to the same Motion Word center in other videos. Even though the environment and the golfers differ, the supervoxel descriptors capture the characteristic motion well, and indeed, the corresponding SVs in the second video also correspond to the golfer's body. Best viewed in color.

future work, these supervoxel features can be the input to methods which model temporal relationships, e.g., [3,11,30,35].

4.4 Datasets

We evaluate the framework on two datasets: the YouTube dataset [23], and the HMDB [17] dataset. The former dataset contains 11 action categories with a total of 1,168 sequences, with roughly 44 test videos per split. It is a challenging dataset due to large variations in camera motion and viewpoint, object appearance, pose, and scale. The HMDB dataset consists of 6849 clips divided into 51 action categories, with 1530 test videos per split. For both datasets, we use the train/test splits provided by the authors.

5 Interpretability

We seek video representations that enable interpretability. That is, when visualized, we want features that give us the power to understand which regions make two videos similar. For instance, given two videos, we can visualize DTs that quantize to the same codebook center. In Figure 4a we select one DT from the region of the golfer's body, find its corresponding codebook center, and then display all DTs that quantize to the same codebook center in two golf videos. Even

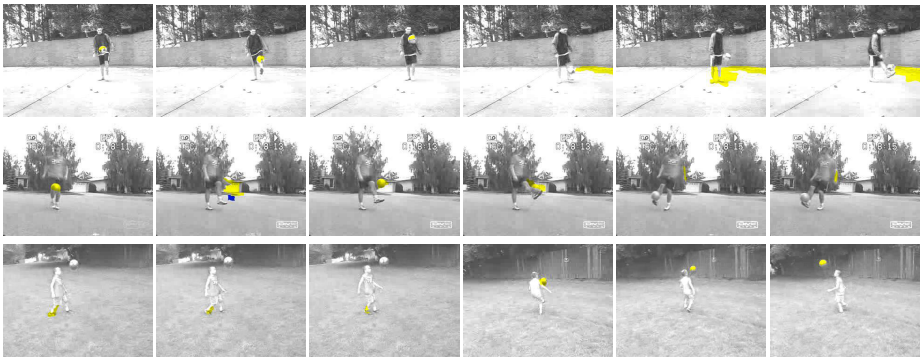


Fig. 6. For the action "juggle," we show some of the unique Motion Words (codebook centers) that are used (expressed) only in this class. They tend to correspond to the soccer ball and the player's legs. Best viewed in color.

Fig. 7. Supervoxels from one of the unique Motion Word centers for the action "horse riding" correspond to regions on the body of the horse when moving to the left. They are found in videos with both slow or fast translations, with or without camera motion. Best viewed in color.

though we can visualize low-level features in this manner, it is difficult to interpret why these DTs have been clustered together. In contrast, since supervoxels are interpretable, the proposed BMW representation enables interpretation of features common across videos. For the same two golf videos, we manually select a supervoxel on the golfer's body in one frame and in Figure 4b we visualize all regions that quantize to the same Motion Word center. We can now easily interpret the similar regions - they indeed correspond to the golfer's body, as expected. Furthermore, the supervoxel based descriptors are robust to environment changes. For a very different golf video, in Figure 5, we visualize supervoxels from from the same codebook center. We find that these regions also correspond to the golfer's body, as desired.

In addition, we can qualitatively evaluate how well the representation captures features specific to each action class. For example, Motion Words that appear only in videos from one action class are unique to that class. The total number of unique Words in the YouTube dataset using the BoW representation is only 83 out of $20,000$ Words, for STP it is 220, and for BMW it is 1280. In Figure 6 we visualize a few of these Motion Words for the action "juggle." The supervoxel

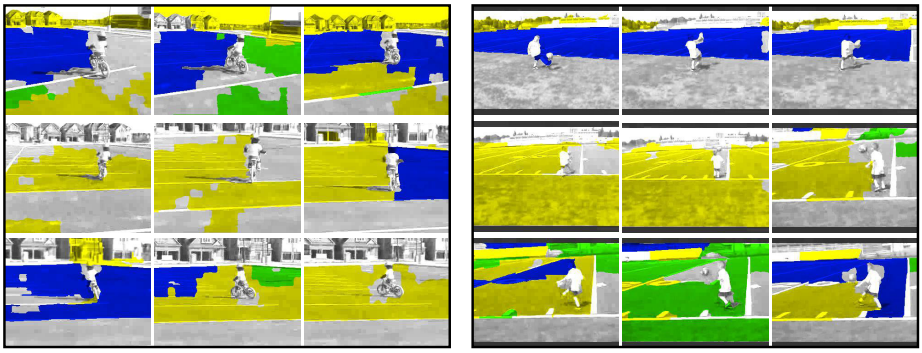


Fig. 8. The query "bike" video (left) is mistakenly classified as "juggle" (right) in a nearest neighbor retrieval task. We can visualize Motion Words that are common between the two videos. Yellow denotes the HOF channel, green the trajectory, and blue denotes multiple channels (including HOG and MBH).

regions roughly correspond to the soccer ball and the legs of the players. In Figure 7 we show a few of the unique Motion Words for the "horse riding" action class, which tend to correspond to the back and legs of the horse.

Furthermore, Motion Words give us the power to understand incorrect results. For example, in a retrieval task with a query "bike" video and a retrieved "juggle" video (Figure 8), we can easily visualize the regions that make the two videos similar in appearance (the soccer field) and the regions with similar motion (the children and cameras in each video move in the same manner).

6 Quantitative Evaluation

6.1 Classification

We learn a one-vs-all SVM [14] classifier with a χ^2 kernel and find the parameters via 5-fold cross validation on the training set. Similarly to Wang *et.al.* [33], we combine the feature channels and report average accuracy[1]. We evaluate the key components of BMW, namely, the supervoxel settings, the low-level features, and the size of the representation.

In Figure 9 we show classification accuracy on the Youtube dataset using Dense Trajectories and different codebook sizes. We compare standard Bag of Words (BoW), Spatio-Temporal Pyramids (STP) [33], three supervoxel methods obtained with the streaming GBH algorithm (GBH coarse, GBH fine, and GBH combined), and supervoxels obtained from the UES algorithm with the "motion-ness" objective. We find that very coarse supervoxels (GBH coarse) are not suitable for pooling dense trajectories. However, the other three supervoxel settings outperform both global pooling and STP. Motion Words based on

[1] We compute average classification accuracy by taking the label of the most confident classification among the one-vs-all SVM classifiers for each test video.

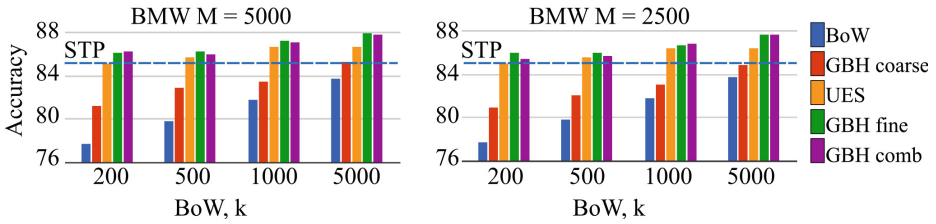


Fig. 9. Classification accuracy on the YouTube dataset using DTs and different codebook sizes. When pooling features over non-coarse supervoxels encoded to large codebooks, we obtain better classification performance compared to BoW and STP.

fine supervoxels improve performance by 3.5% (88.9) compared to coarse spatio-temporal pyramids (85.4 [34]), and similar performance to the recent Fisher vector (MBH and SIFT) representation pooled over STPs (89 [28]). Fine supervoxels better capture motion information compared to coarse supervoxels, which often group regions of different motion and appearance. We observe the same trend when learning Motion Word codebooks of sizes 5000 and 1000, showing that the performance of the proposed representation is not very sensitive to M.

Table 1. Pooling different types of low-level descriptors in BMW compared to standard pooling in the YouTube dataset (5000 codebook centers). DTs pooled over non-coarse supervoxels achieve highest classification accuracy.

	BoW	GBH coarse	GBH fine	GBH comb	UES
ISA [19]	75.8	75.9	76.4	77.2	76.1
Dense HOG,HOF,MBH [33]	81.4	82.8	85.7	85.8	83.6
Dense Trajectories [33]	**83.8**	**85.3**	**88.9**	**88.1**	**86.8**

Table 2. On the YouTube dataset, the proposed BMW achieves classification accuracy comparable to state-of-the-art methods, while enabling interpretation and clear visualization of the video representation

BoW	STP [33]	FV MBH+SIFT STP [28]	FV MBH STP [28]	BMW GBH fine
83.8	85.4	89	88.5	88.9

Next, using 5000 codebook centers, we evaluate performance of the underlying low-level features. In Table 1 we show classification accuracy on the YouTube dataset using the proposed Motion Words representation where we pool different types of low-level descriptors: DTs [33], which capture HOG, HOF, and MBH over 15 frames by tracking interest points; Dense Descriptors (HOG, HOF, MBH) [33] which do not track points; and automatically learned ISA features computed over cuboids. Compared to the standard BoW representation, pooling over supervoxels always performs better. The Dense Trajectories capture

temporal information better than the Dense Descriptors and obtain higher performance. We find that the ISA features are not suitable for pooling over supervoxels since they are computed over cuboids of sizes much larger than the extracted supervoxels. Highest performance is achieved when pooling features that encode temporal information (DTs) over fine supervoxels (which robustly group pixels of similar motion and appearance).

Finally, we evaluate the BMW representation on the challenging HMDB dataset. In Table 3 we show classification performance compared to prior results reported by Wang *et.al.* [33,34] and Oneata *et.al.* [28]. Wang *et.al.* [33,34] use DTs with combined HOG, HOF and MBH channels, and pool over spatio-temporal pyramids. The latter work augments DTs to compensate for camera motion. Oneata *et.al.* [28] extract spatial Fisher vectors based on MBH and SIFT descriptors, pooling over spatio-temporal grids. We only evaluate non-coarse supervoxels using DTs and the combined HOG, HOF and MBH channels, learning codebooks of size 5000 centers. The proposed representation achieves state-of-the-art results, while enabling interpretability. We attribute the good performance of the fine supervoxels to the ability of the segmentation method to respect motion boundaries. BMW with fine supervoxels obtains 58.8% classification accuracy, which is 2.6% better than the 57.2% previously reported by Wang *et.al.* [34].

Table 3. Classification accuracy on the HMDB dataset using BMW with fine GBH and UES supervoxels and codebooks of size 5000 centers. Pooling over fine supervoxels obtains better performance compared to other pooling methods.

	Wang *et.al.* [33]	Oneata *et.al.* [28]	Wang *et.al.* [34]	GBH fine	UES
HMDB	48.3	54.8	57.2	**58.8**	57.9

Table 4. Nearest neighbor retrieval (average recall) on the YouTube dataset. When pooling DTs over non-coarse supervoxels, the Motion Words representation outperforms global and STP pooling methods for different codebook sizes.

	BoW	STP	GBH coarse	UES motion	GBH fine	GBH comb
$K = 500$	65.43	66.80	67.21	68.23	68.35	**68.69**
$K = 1000$	67.94	68.85	68.31	69.14	69.53	**69.59**
$K = 5000$	68.22	68.70	67.61	69.14	69.34	**69.52**

6.2 Retrieval

We analyze the usefulness of the proposed representation in the task of directly comparing videos using nearest neighbor. We simply concatenate the descriptors for each video, and treat the test set as query videos. In Table 4 we report average recall from a nearest neighbor retrieval task on the YouTube dataset using χ^2 distance, where we use the provided action labels to determined correct retrieval.

Similarly to the classification performance results, we find that coarse supervoxels do not provide good spatio-temporal support for pooling low-level features. However, we find that pooling over the other supervoxels settings outperforms BoW and STP representations in this very challenging task.

7 Conclusion

In this paper we propose Motion Words – a representation that builds upon BoW by pooling features over supervoxels and performing a second quantization step to obtain a robust and compact video representation. We show that this representation is well-suited for activity classification and retrieval, achieving state-of-the-art performance. The BMW representation achieves high performance when we encode non-coarse supervoxels to BoW codebooks of Dense Trajectories. Furthermore, Motion Words enable interpretability of the results, giving us the power to gain understanding of which features make videos similar.

References

1. Bettadapura, V., Schindler, G., Ploetz, T., Essa, I.: Augmenting bag-of-words: Data-driven discovery of temporal and structural information for activity recognition. In: CVPR (2013)
2. Boureau, Y.L., Le Roux, N., Bach, F., Ponce, J., LeCun, Y.: Ask the locals: Multiway local pooling for image recognition. In: ICCV, pp. 2651–2658 (2011)
3. Brendel, W., Todorovic, S.: Video object segmentation by tracking regions. In: ICCV, pp. 833–840 (2009)
4. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 282–295. Springer, Heidelberg (2010)
5. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: CVPR, pp. 60–65 (2005)
6. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic Segmentation with Second-Order Pooling. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 430–443. Springer, Heidelberg (2012)
7. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising with block-matching and 3D filtering. In: Electronic Imaging (2006)
8. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (2005)
9. Everts, I., van Gemert, J.C., Gevers, T.: Evaluation of color STIPs for human action recognition. In: CVPR (2013)
10. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2), 167–181 (2004)
11. Gaidon, A., Harchaoui, Z., Schmid, C.: Actom Sequence Models for Efficient Action Detection. In: CVPR, pp. 3201–3208 (2011)
12. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV, pp. 1–8 (2009)

13. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation, In: CVPR (2010)
14. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Tech. rep., National Taiwan University (2005)
15. Jain, A., Gupta, A., Rodriguez, M., Davis, L.S.: Representing Videos using Mid-level Discriminative Patches. In: CVPR (2013)
16. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: CVPR, pp. 3304–3311 (2010)
17. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: ICCV (2011)
18. Laptev, I., Lindeberg, T.: Local descriptors for spatio-temporal recognition. In: MacLean, W.J. (ed.) SCVMA 2004. LNCS, vol. 3667, pp. 91–103. Springer, Heidelberg (2006)
19. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR (2011)
20. Lee, Y.J., Kim, J., Grauman, K.: Key-segments for video object segmentation. In: ICCV (2011)
21. Lezama, J., Alahari, K., Sivic, J., Laptev, I.: Track to the future: Spatio-temporal video segmentation with long-range motion cues. In: CVPR (2011)
22. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: CVPR (2011)
23. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: CVPR (2009)
24. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: ICCV, pp. 2272–2279 (2009)
25. Mathe, S., Sminchisescu, C.: Dynamic Eye Movement Datasets and Learnt Saliency Models for Visual Action Recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 842–856. Springer, Heidelberg (2012)
26. Moore, A., Prince, S., Warrell, J., Mohammed, U., Jones, G.: Superpixel lattices. In: CVPR (2008)
27. Nguyen, M.H., Torresani, L., De la Torre, F., Rother, C.: Weakly supervised discriminative localization and classification: a joint learning process. Tech. rep., Carnegie Mellon University (2009)
28. Oneata, D., Verbeek, J., Schmid, C.: Action and Event Recognition with Fisher Vectors on a Compact Feature Set. In: ICCV, pp. 1817–1824 (2013)
29. Raptis, M., Kokkinos, I., Soatto, S.: Discovering discriminative action parts from mid-level video representations. In: CVPR (2012)
30. Shapovalova, N., Vahdat, A., Cannons, K., Lan, T., Mori, G.: Similarity constrained latent support vector machine: An application to weakly supervised action classification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 55–68. Springer, Heidelberg (2012)
31. Tighe, J., Lazebnik, S.: Superparsing: Scalable nonparametric image parsing with superpixels. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 352–365. Springer, Heidelberg (2010)
32. Vazquez-Reina, A., Avidan, S., Pfister, H., Miller, E.: Multiple hypothesis video segmentation from superpixel flows. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 268–281. Springer, Heidelberg (2010)

33. Wang, H., Kläser, A., Schmid, C., Cheng-Lin, L.: Action Recognition by Dense Trajectories. In: CVPR (2011)
34. Wang, H., Schmid, C.: Action Recognition with Improved Trajectories. In: ICCV (2013)
35. Wang, L., Qiao, Y., Tang, X.: Motionlets: Mid-level 3D parts for human motion recognition. In: CVPR (2013)
36. Xu, C., Whitt, S., Corso, J.: Flattening supervoxel hierarchies by the uniform entropy slice. In: ICCV (2013)
37. Xu, C., Xiong, C., Corso, J.J.: Streaming hierarchical video segmentation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 626–639. Springer, Heidelberg (2012)
38. Zhang, Y., Liu, X., Chang, M.-C., Ge, W., Chen, T.: Spatio-temporal phrases for activity recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 707–721. Springer, Heidelberg (2012)
39. Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image classification using super-vector coding of local image descriptors. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 141–154. Springer, Heidelberg (2010)
40. Zhu, Y., Nayak, N.M., Roy-Chowdhury, A.K.: Context-aware modeling and recognition of activities in video. In: CVPR (2013)