Angelini, F., Fu, Z., Velastin, S.A, Chambers, J.A. y Naqvi, S.M. (2018). 3D-Hog Embedding Frameworks for Single and Multi-Viewpoints Action Recognition Based on Human Silhouettes. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

DOI: https://doi.org/10.1109/ICASSP.2018.8461472

3D-HOG EMBEDDING FRAMEWORKS FOR SINGLE AND MULTI-VIEWPOINTS ACTION RECOGNITION BASED ON HUMAN SILHOUETTES

Federico Angelini[⋆] *Zeyu Fu*[⋆] *Sergio A. Velastin*[†‡] *Jonathon A. Chambers*[⋆] *Syed Mohsen Naqvi*[⋆]

[⋆]Intelligent Sensing and Communications Research Group, Newcastle University, UK
[†]Department of Computer Science and Engineering, University Carlos III de Madrid, Spain
[‡]School of EECS, Queen Mary University of London, UK

ABSTRACT

Given the high demand for automated systems for human action recognition, great efforts have been undertaken in recent decades to progress the field. In this paper, we present frameworks for single and multi-viewpoints action recognition based on Space-Time Volume (STV) of human silhouettes and 3D-Histogram of Oriented Gradient (3D-HOG) embedding. We exploit fast-computational approaches involving Principal Component Analysis (PCA) over the local feature spaces for compactly describing actions as combinations of local gestures and L_2-Regularized Logistic Regression (L_2-RLR) for learning the action model from local features. Outperforming results on Weizmann and i3DPost datasets confirm efficacy of the proposed approaches as compared to the baseline method and other works, in terms of accuracy and robustness to appearance changes.

***Index Terms*—** Single-viewpoint, Multi-viewpoints, Human Action Recognition, 3D Histogram of Oriented Gradient, Silhouettes

1. INTRODUCTION

Human action recognition is a challenging and growing field. Applications are mainly related to surveillance, video compression, sport analysis, interfaces design and gaming [1]. In the last few years, many efforts have been performed to enhance discriminative power, robustness to appearance, viewpoints changes, occlusions and limiting the computational costs. However, the state-of-art is still far from being reliable, real-time and robust, especially in scenarios with moving backgrounds, partially occluded human body shapes and with many interacting subjects.

In this paper, we present frameworks based on 3D-HOG embedding [2] for applications in static background scenarios where human silhouettes are available. Our contribution is twofold. First, the proposed methods are faster, more accurate and stable than the baseline in [2], providing a solution for local descriptor combination in the form of a short-length features vector. Second, we outperform multi-viewpoints state-of-art recognition accuracies and robustness to appearance changes over the tested datasets.

Referring to the taxonomy suggested by [1], a common procedure for multi-viewpoints action recognition is to learn models from view-independent features [3]. However, this needs to find common features among videos from different point of views. Reversely, finding peculiar features into samples from a fixed point of view is the key of the proposed method. Therefore, labelling action samples from different point of views with the same label, we can learn a generalised multi-viewpoints model.

Many works have been presented relying on *histogram of oriented gradient* (HOG) [4–7]. However, all of them are based on 2D-HOG features, because the action is practically stored as *motion-history* image. In this paper, following the idea originally proposed in [2], we exploit the potentialities of *Space-Time Volumes* (STV) for representing an action and 3D-HOG features. We use the same approach regarding the embedding of features with a prototypes library, to convert features into fixed-length descriptors insensitive to the time-length of the analised action. However, from this stage onward, the proposed frameworks take a different direction to deal with these descriptors and to perform the recognition task. While the original approach was based on unrelated local decisions to take a final decision, our approaches rely on local searching of *gestures* to learn a cross-location model for action recognition. These approaches are expected to allow more coherence in the decision strategy, as well as to be more effective in terms of accuracy. It turns out that, among these advantages, the proposed frameworks also reduce the computational effort, achieving improved performance by using a smaller prototypes library.

We used the Weizmann and i3DPost datasets to compare the proposed approaches and the baseline performances. Both datasets have been also useful for establishing comparison with other methods, in single-viewpoint and multi-viewpoints setting, and to test the robustness to appearance changes. Despite the Weizmann dataset being a *closed problem* (as some authors have presented perfect accuracy results), it is currently used as testbed for silhouette-based methods.

In Section 2, we will summarise the 3D-HOG embedding algorithm to extract features from each STV and how the authors in [2] have used it to learn a model for action recognition. This will be the *baseline* for the proposed method presented in Section 3, which can be then intended as an optimised variation of the 3D-HOG embedding algorithm [2]. In Section 4, we present results on Weizmann and i3DPost datasets, reporting evidences of improvements with respect to the baseline and other approaches. In Section 5, conclusions of this work are summarised.

2. PRELIMINARIES

2.1. Problem statement

Let $\mathbb{D} = \{s_i, l_i, w_i\}_{i=1}^{N}$ be the multi-viewpoints action dataset containing N samples, where s_i is the ith RGB video sample, $l_i \in \mathcal{L} = \{1, \ldots, L\}$ is the ith action label where L is the whole number of considered action labels, $w_i \in \mathcal{W} = \{1, \ldots, W\}$ the ith point of view label and W the whole number of considered point of views. Let $\mathbb{T} \subset \mathbb{D}$ be the chosen training subset and $\mathbb{T}^* = \mathbb{D} \backslash \mathbb{T}$ the testing subset. The aim is to extract features from each sequence $s_i \in \mathbb{D}$ to learn a model by using $\mathbb{T}$ to recognise actions sequences in $\mathbb{T}^*$.

It is obvious that when $W = 1$, $\mathbb{D}$ reduces to be a single-

viewpoint dataset.

We need s_i to be in the form of a 3D matrix obtained by concatenating binary masks for image sequences of the ROI. We will refer to s_i as a *Space-Time Volume* (STV) as defined in [1] and it is shown in Fig. 1.

2.2. 3D-HOG feature extraction

STVs are partitioned into overlapped blocks of a fixed dimension. Binary data within each block are then used to compute the 3D-HOG descriptor as suggested in [2], exploiting the 3D vectorial gradients field. Thus, each block remains associated with a vector $\mathbf{b}$ obtained with the concatenation of SIFT-like [8] histogram of oriented gradients. This descriptor can be considered highly informative with respect to the 3D-shape defined by binary data within the block [2], as well as robust to noise and small data deformations.

2.2.1. Prototypes library embedding

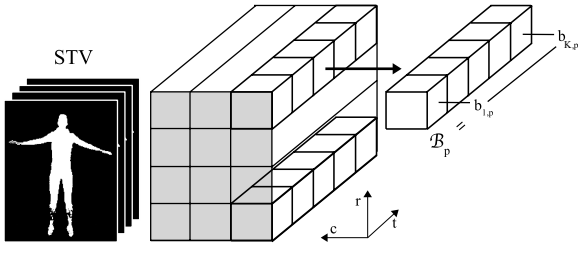

Fig. 1. STV and its overlapped blocks partitioning. The flow $\mathcal{B}_p$ is depicted, for a generic position $p = (r_0, c_0)$.

The block descriptors are then organised as follows: for all locations $p = (r_0, c_0)$ in the STV, we consider the *flow at location* p, $\mathcal{B}_p = \{\mathbf{b}_{1,p}, \ldots, \mathbf{b}_{K,p}\}$, where $\mathbf{b}_{i,p}$ is the ith block descriptor at location p and K is the total number of block descriptors along time. We can consider different locations p depending on the chosen space partition. Without loss of generality, let $p \in \{1, \ldots, P\}$, where P is the number of considered location in the (r, c) plane. In Fig. 1, a global perspective of STV and blocks partitioning have been shown.

Following the strategy in [2], for a fixed label l and point of view label w, we *randomly* choose n block descriptors within the training STVs. This will lead to a descriptors library V such that $|V| = nLW$. Without loss of generality, we can define $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_{nLW}\}$, containing n descriptor prototypes for each action and each point of view included in the dataset.

Therefore, we compute the *embedding* $D_i(\mathcal{B}_p)$ of each block flow $\mathcal{B}_p$ with the library V as follows:

$$D_i(\mathcal{B}_p) = \min_{j=1,\ldots,K} d(\mathbf{b}_{j,p}, \mathbf{v}_i) \qquad i = 1, \ldots, nLW \tag{1}$$

where d represents a convenient distance, i.e. *Euclidean* or χ^2 distance. Thus, the *embedded vector* $\mathcal{D}_p$ at location p is defined as

$$\mathcal{D}_p = [D_1(\mathcal{B}_p), \ldots, D_{nL}(\mathcal{B}_p)] \tag{2}$$

It is worth underlining that the prototypes library V contains prototypes randomly chosen within training sequences *regardless of the location* p. Thus, the training sequences are embedded against V producing some zero-entries in $\mathcal{D}_p$ when $\mathcal{B}_p$ contains prototypes chosen to belong to the library V.

2.2.2. Final flow-based decision

For each location p, a model is learned by using training embedded vectors $\mathcal{D}_p$. In particular, by using L_2-RLR [9] we can estimate the probability $\mathbb{P}(l_p^* = l|\mathcal{D}_p^*, \Theta_p)$, of a testing embedded vector $\mathcal{D}_p^*$ to belong to one of the considered action classes l, where Θ_p represents the positional learned model.

Depending on the chosen space partition, this strategy leads to *flow-based* independent decisions in number of P. To combine all these decisions to classify the action in the STV, we report here the *Sum Rule* [2] being considered at one time easy to implement, effective and comparable in terms of results to other strategies. The *Sum Rule* consists of selecting the final label l^* such that

$$l^* = \max_{l \in \mathcal{L}} \sum_{p=1}^{P} \mathbb{P}(l_p^* = l|\mathcal{D}_p^*, \Theta_p) \tag{3}$$

3. PROPOSED FRAMEWORKS

3.1. Robust prototypes selection

We replaced the cross-location random selection of prototypes with a novel location-based selection strategy. If H is the total number of descriptors with the same label l from the same point of view w in $\mathbb{T}$ at location p, we can collect the block descriptors $\{\mathbf{b}_{j,p}\}_{j=1}^{H}$. They can be seen as a set of points within a multidimensional cartesian space. By using the *hierarchical clustering algorithm* [10] we can look for n groups of descriptors for each action and each point of view, that is

$$\{\mathbf{b}_{j,p}\}_{j=1}^{H} = \{\mathbf{b}_{j,p}\}_{j=1}^{S_1} \cup \{\mathbf{b}_{j,p}\}_{j=S_1}^{S_2} \cup \cdots \cup \{\mathbf{b}_{j,p}\}_{j=S_{n-1}}^{S_n} \tag{4}$$

where S_1 is the cardinality of the first group, $S_2 - S_1$ the cardinality of the second group and similarly $S_n - S_{n-1}$ is the cardinality of the last group. In (4), action and point of view symbols are implicit, as all descriptors have fixed l and w. Thus, by averaging elements within the same group, n new prototypes remain defined. Considering all labels l and w, this leads to a new library V such that $|V| = nLW$ prototypes, no longer necessarily included within the training subsequences. Thus, the embedded vectors are defined by using (1) and (2). As a consequence of this library definition, the embedded vectors will have no zero-entries, helping the subsequent learning task to avoid overfitting. Moreover, as we will see in Section 4, this deterministic strategy stabilises the whole process.

3.2. Overcoming of flow-based decisions rules

We propose two novel approaches for combining local embedded vectors for a cross-position classification. It turns out that the first approach is suitable in presence of single-viewpoint datasets. Instead, the second one is more suitable when multi-viewpoints samples are available. It is worth underlining that the single-viewpoint datasets have in general fewer samples to characterise an action, not having samples from different point of views, which makes the training process remarkably different to the multi-viewpoints one.

3.2.1. Single-viewpoint method

For a fixed location p, the training embedded vectors can be seen as points within a multidimensional space. Thus, by exploiting only training data labels l, we can perform the following *training* process:

1. Exploiting training embedded vectors $\mathcal{D}_p$, we can apply the L_2-RLR and learn a positional model Θ_p;

2. Exploiting training embedded vectors $\mathcal{D}_p$, we can also compute the center class points matrix $\mathbf{C}_p = \{\mathbf{c}_{p,l}\}_{l\in\mathcal{L}}$;
3. Performing PCA over $\mathbf{C}_p$ selecting a small and fixed number of components α_1, we obtain $\bar{\mathbf{C}}_p = \{\bar{\mathbf{c}}_{p,l}\}_{l\in\mathcal{L}}$;

Therefore, given $\bar{\mathbf{C}}_p = \{\bar{\mathbf{c}}_{p,l}\}_{l\in\mathcal{L}}$ for all p, we can concatenate them along p, that is $\bar{\mathcal{C}} = \{[\mathbf{c}_{1,l}, \ldots, \mathbf{c}_{P,l}]\}_{l\in\mathcal{L}}$ which supplies a new learning process based on L_2-RLR, exploiting training data, to get a cross-locations model Θ.

A testing embedded vector $\mathcal{D}_p^*$ is labelled with a positional label l_p^* by using the learnt model Θ_p and associated with $\mathbf{c}_{p,l_p^*}$. Thus, the testing sample will be associated with a single vector given by concatenating $\mathbf{c}_{p,l_p^*}$ for all p, that is $\bar{\mathcal{D}}^* = [\mathbf{c}_{1,l_1^*}, \ldots, \mathbf{c}_{P,l_P^*}]$.

The final decision l^* is obtained by using Θ to test $\bar{\mathcal{D}}^*$. The parameter α_1 in the training process can be established via cross-validation.

3.2.2. Multi-viewpoints method

Fixed the location p, the *training* process can be reduced to a PCA over the training embedded vectors $\{\mathcal{D}_{p,i}\}_{i=1}^{|\mathbb{T}|}$, considering the number of principal components that reaches the cumulative α_2 percentage of explained variance. Let $\{\bar{\mathcal{D}}_{p,i}\}_{i=1}^{|\mathbb{T}|}$ be the transformed training embedded vectors and $\mathbf{A}_p$ the transformation matrix. Therefore, we can center and transform the embedded testing vector $\mathcal{D}_p^*$ by using $\mathbf{A}_p$, thus

$$\mathbf{A}_p \left(\mathcal{D}_p^* - \frac{1}{|\mathbb{T}|} \sum_{i=1}^{|\mathbb{T}|} \mathcal{D}_{p,i} \right) = \bar{\mathcal{D}}_p^* \tag{5}$$

Therefore, each training sample s_i will be associated with a single vector given by concatenating $\bar{\mathcal{D}}_{p,i}$, for all p. Similarly, testing sample will be associated with a single vector given by concatenating $\bar{\mathcal{D}}_p^*$ for all p. These can supply an L_2-RLR, exploiting training data labels l, to get the final label l^*. The parameter α_2 in the training process can be established via cross-validation.

4. EXPERIMENTS

The single-viewpoint Weizmann dataset consists of video samples of 10 actions performed by 9 actors recorded from single action-related point of views. Binary masks are publicly available in the requested STV form. Regarding the multi-viewpoints i3DPost dataset, it consists of 6 single-actor actions videos, 2 multi-actors actions videos, 4 multi-actions single-actor videos and facial-expressions data. The dataset is performed by 8 actors and recorded from 8 different point of views. We selected the whole dataset, ignoring videos containing multiple-actions and facial-expressions. We relied on the ViBE algorithm [11] for background subtraction, to get the binary masks of the scene. Then, the ROIs around the subjects are hand-picked exploiting the human shape centroid.

As in [2], for both datasets, the STVs have been rescaled to $64 \times 48 \times t$ pixels. The block dimension is fixed to be $16 \times 16 \times 16$ pixels, with an overlapping of 8 pixels. For the 3D-HOG feature extraction, we used the setting in [2].

The robustness to human appearance changes has been tested with the *leave-one-actor-out* (LOAO) experimental setting. Thus, one actor samples are kept out from the training and used for testing. Accuracy results are given on average over all possible LOAO configurations.

We report experiments where $n \in \{5, 10, 15, 20, 30, 40\}$ because no significant advantages in terms of accuracy have been achieved by using more computational expensive higher values.

The averaged accuracy results for our implementation of the baseline [2] are provided with standard deviation σ, over ten different choices of the random prototypes in V. Regarding the proposed method, since it is based on a deterministic strategy for choosing prototypes in V, the results are fixed for each n, without any variation.

4.1. Single-viewpoint results (LOAO)

Regarding Weizmann dataset, our implementation of the baseline method has confirmed the perfect result (accuracy 100%) reported in [2] for 9 actions over 10 (WAVE2 out), by using $n = 30$ prototypes per action. However, [2] does not provide results for the whole set of actions of the Weizmann dataset. Our implementation of [2] shows an accuracy of at most 98.66% with $n = 30$ with a standard deviation $\sigma = 0.61$. The proposed framework is able to achieve stably 100% of accuracy with $n = 20$ and $\alpha_1 = 6$. This is among the best results reported so far for Weizmann dataset. Fig. 2(a) provides comparisons between the proposed framework and the baseline results in the 10-actions setting for different values of n. Table 1 shows comparisons between the proposed framework and other approaches.

Regarding i3DPost, despite it is a multi-viewpoints dataset, it can also be used for single-viewpoint experiments, passing to the training and testing process only samples from a fixed point of views. Thus, the following results are given on average over the eight point of views. Our implementation of [2] achieves 97.46% accuracy with $n = 40$. Instead, with only $n = 10$ ($\sigma_1 = 6$), the proposed framework achieves 98.24% accuracy. Fig. 2(b) shows comparisons between the proposed framework and the baseline results in the 8-actions setting for different values of n.

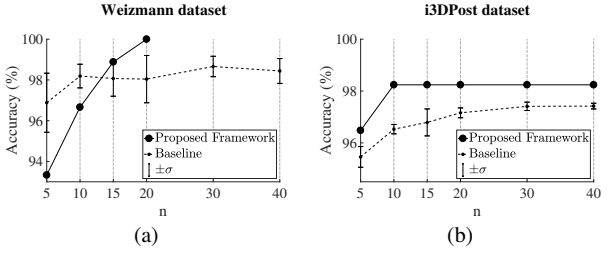

Fig. 2. Relations between n and accuracy for the proposed method and the baseline (LOAO, single-viewpoint setting in Section 4.1, $\alpha_1 = 6$). (a) Results for Weizmann dataset with 10 actions. (b) Results for i3DPost dataset with 8 actions.

4.2. Multi-viewpoints results

Our implementation of the baseline method in the 8-actions and 8-viewpoints setting achieves at most 98.86% accuracy with $n = 30$ and standard deviation $\sigma = 0.17$. However, the proposed method achieves a higher and stable accuracy of 99.60% with $n = 30$ and $\sigma_2 = 95\%$ in the same setting. We also tested the 6-actions setting (neglecting interaction samples) achieving an accuracy of 99.73% with $n = 30$ and $\sigma_2 = 99\%$.

Comparisons with other state-of-arts results on the i3DPost dataset

Method	L	Accuracy	n	α_1
Proposed Framework	10	**100%**	20	6
Gorelick et al. [12]	10	**100%**	-	-
Jiang et al. [13]	10	**100%**	-	-
C.Li et al. [5]	9	97.53%	-	-
Ahsan et al. [14]	9	97.5%	-	-
Ahsan et al. [14]	10	94.26%	-	-

Table 1. Comparisons for Weizmann dataset (LOAO).

Method	L	W	Accuracy	n	α_2
Prop. Framework ⊛	8	8	**99.60%**	30	95%
Prop. Framework ⊛	6	8	**99.73%**	30	99%
Castro et al. [16] ⊙	6	2	99.00%	-	-
Iosifidis et al. [17]	6	8	98.16%	-	-
Iosifidis et al. [17]	8	8	96.34%	-	-
Azary et al. [18]	6	8	92.97%	-	-
Hilsenbeck et al. [15] ⊙⊛	6	8	92.42%	-	-

Table 2. Accuracy results for i3DPost dataset (LOAO). The ⊙ highlights methods with automatic selection of ROIs, while ⊛ highlights those with automatic background subtraction methods without prior knowledge.

with LOAO setting can be done as long as pre-processing steps are considered, such as ROI detection and background subtraction. However, no common evaluation protocol has been fixed for this dataset in the literature. Despite this, inspired by the comparison approach suggested in [15], we report in Table 2 the best results reported in literature with LOAO setting to the best of our knowledge. Methods where the pre-processing steps are entirely entrusted to the machine are highlighted. We underline that the length of concatenated vectors in Section 3.2.2 is 5145 ($L = 8$) and 7943 ($L = 6$), which can be considered as compressed action expressions.

4.3. Computational complexity

In the baseline and proposed framework, the embedding is the most expensive task, depending on the time-length K of the samples, the number of actions L and on the size n of the library V. Moreover, it is necessary in both training and testing phases. For each location p, the embedding vector $\mathcal{D}_p$ is composed of nLW entries, each of them computed as a minimisation after a one-to-many comparison. In formulas, we can express the complexity of the embedding procedure with respect to n as $f(n) = (c_1 + Kc_2 + \mathcal{O}(K))nLW = \mathcal{O}(n)$, where c_1, c_2 and c_3 are positive constants and where we have assumed that the complexity for the minimisation problem in (1) is $\mathcal{O}(K)$. This shows that choosing n as low as possible is important for fast-computation applications. Results in Sections 4.1 & 4.2 show that our framework achieves better performance in terms of accuracy for equivalent n than the baseline. Thus, it is preferable to the baseline in real-time oriented implementations.

4.4. Discussion

Our implementation of the baseline method shows a drawback regarding the selection of prototypes in V, which is cross-location and overfitting prone. Therefore, the final accuracy results are considerably sensible to the random selection of prototypes per action n to build up the library V. Only by sufficiently increasing n to high levels this variability is partially mitigated, albeit at the price of more computational effort. Moreover, to reach acceptable levels of accuracy, it is necessary to increase n up to levels where the computational cost of the embedding procedure affects the execution time, compromising real-time applications. However, the proposed framework is able to increase the performance of the baseline method, decreasing the necessary n to reach higher accuracy rates, combining local descriptors into a short-length feature vector which can supply subsequent analysis.

These promising results rely on a more sophisticated method to manage flows information. The baseline method is based only on flows and action labels l, on the assumption that in each flow different actions are actually recognisable. However, within a certain flow at location p, not all the actions can be distinguishable, implying mistakes and misclassifications in the learning stage. A certain (a-priori unknown) number of simple *gestures* are expressed by the embedded vectors. For example, focusing on two actions like WAVE-ONE-HAND and WAVE-TWO-HANDS in Fig. 3, it is clear that those flows where local gestures look similar will carry the same information. Thus, an action-label-based approach (such as the baseline method) over those flows is destined to fail. On the contrary, the proposed methods determine the *local gestures composition* to discriminate between actions, exploiting PCA as *continuous multidimensional clustering* method [19]. Thus, even if most of the flows are capturing the same gestures, showing after PCA similar continuous multidimensional local labels, the final learning process can rely on those flows where the gestures are dissimilar.

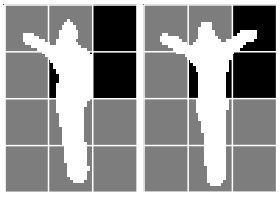

Fig. 3. Example of STVs of two actions (WAVE-ONE-HAND and WAVE-TWO-HANDS). Blocks showing the same local gestures (grey blocks) and blocks showing significant differences (black blocks).

5. CONCLUSIONS

In this paper, we presented frameworks for silhouette-based human action recognition based on 3D-HOG for Space-Time Volume actions representation, which is as a variation of the 3D-HOG embedding algorithm [2].

We achieved better performance in terms of accuracy, computational effort and stability over the tested datasets than the baseline method, providing a solution for combining local descriptors into a single short-length features vector. Outperforming results with respect to other works with similar experimental settings have been shown.

Future works will focus upon testing the proposed framework over other datasets, incorporating automatic ROI selection systems. Moreover, evaluations of robustness to occlusions and viewpoint changes will be conducted. We are also interested in overcoming the inherent rigidity of flows-based methods involving pose-based methods such [20], to combine different features extraction methods improving the discriminative power and the overall robustness.

6. REFERENCES

[1] J. K. Aggarwal, "Human Activity Analysis: A Review," *ACM Comput. Surv. Article*, vol. 43, no. 3, pp. 1–43, 2011.

[2] D. Weinland, M. Özuysal, and P. Fua, "Making action recognition robust to occlusions and viewpoint changes," in *LNCS - Lecture Notes in Computer Science*, vol. 6313, no. part 3, 2010, pp. 635–648.

[3] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE CVPR*, vol. I, 2005, pp. 886–893.

[5] C. Li, Y. Liu, J. Wang, and H. Wang, "Combining Localized Oriented Rectangles and Motion History Image for Human Action Recognition," in *ISCID*, 2014, pp. 53–56.

[6] F. Liu, X. Xu, S. Qiu, and C. Qing, "Simple to Complex Transfer Learning for Action Recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 949–960, 2016.

[7] F. Murtaza, M. H. Yousaf, and S. A. Velastin, "Multi-view human action recognition using 2D motion templates based on MHIs and their HOG description," *IET Computer Vision*, vol. 10, no. 7, pp. 758–767, 2016.

[8] D. G. Lowe, "Distinctive image features from scale invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[10] B. K. Lavine, "Clustering and classification of analytical data," in *Encyclopedia of Analytical Chemistry*, 2000, pp. 1–21.

[11] O. Barnich and M. V. Droogenbroeck, "ViBe : A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.

[12] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[13] Z. Jiang, Z. Lin, and L. Davis, "Recognizing actions by shape-motion prototype trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 533–547, 2012.

[14] S. M. M. Ahsan, J. K. Tan, H. Kim, and S. Ishikawa, "Histogram of spatio temporal local binary patterns for human action recognition," in *SCIS*, 2014, pp. 1007–1011.

[15] B. Hilsenbeck, D. Munch, H. Kieritz, W. Hubner, and M. Arens, "Hierarchical Hough forests for view-independent action recognition," in *ICPR*, 2016, pp. 1911–1916.

[16] G. Castro-Munoz and J. Martinez-Carballido, "Real Time Human Action Recognition Using Full and Ultra High Definition Video," in *CSCI*, 2015, pp. 509–514.

[17] A. Iosifidis, A. Tefas, and I. Pitas, "Multi-view action recognition based on action volumes, fuzzy distances and cluster discriminant analysis," *Signal Processing*, vol. 93, no. 6, pp. 1445–1457, 2013.

[18] S. Azary and A. Savakis, "Multi-view action classification using sparse representations on Motion History Images," in *2012 Western New York Image Processing Workshop*, 2012, pp. 5–8.

[19] C. Ding and X. He, "K-means clustering via principal component analysis," *Proceedings of the twenty-first international conference on Machine learning*, vol. Cl, no. 2000, p. 29, 2004.

[20] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in *CVPR*, 2017, pp. 7291–7299.