Ordered Trajectories for Large Scale Human Action Recognition

O. V. Ramana Murthy[1]
[1]Vision & Sensing, HCC Lab,
ESTeM, University of Canberra
O.V.RamanaMurthy@ieee.org

Roland Goecke[1,2]
[2]IHCC, RSCS, CECS,
Australian National University
roland.goecke@ieee.org

Abstract

Recently, a video representation based on dense trajectories has been shown to outperform other human action recognition methods on several benchmark datasets. In dense trajectories, points are sampled at uniform intervals in space and time and then tracked using a dense optical flow field. The uniform sampling does not discriminate objects of interest from the background or other objects. Consequently, a lot of information is accumulated, which actually may not be useful. Sometimes, this unwanted information may bias the learning process if its content is much larger than the information of the principal object(s) of interest. This can especially escalate when more and more data is accumulated due to an increase in the number of action classes or the computation of dense trajectories at different scales in space and time, as in the Spatio-Temporal Pyramidal approach. In contrast, we propose a technique that selects only a few dense trajectories and then generates a new set of trajectories termed 'ordered trajectories'. We evaluate our technique on the complex benchmark HMDB51, UCF50 and UCF101 datasets containing 50 or more action classes and observe improved performance in terms of recognition rates and removal of background clutter at a lower computational cost.

1. Introduction

Human action recognition requires the modelling of the coordinated motions – the actions – of different parts of the human body, their interaction with other objects or persons nearby, and their classification. In early research, human action recognition focussed on classifying only very few $(6-11)$ action classes, which mostly involved motion of the whole body without much interaction with nearby objects or people. Experimental validations were mainly carried out on videos collected in very controlled environments, such as in a laboratory, staged by a single person on several occasions. However, with the widespread use of the internet these days, users are uploading millions of videos on social networking sites such as YouTube and Facebook. This has created challenges for developing robust techniques for action recognition in real-world videos with a large number of action classes. Real-world videos can contain movements of the entire body or of only some specific regions, e.g. facial expressions or moving a limb, possibly repetitive, whole-body movements such as walking and running, or a number of sequences of body movements such as walking in a queue or cross-walking at an intersection. It is of interest to investigate how to adapt, generalise or fuse the existing techniques to model any kind of human actions. In real-world videos, context (= the environment / situation) and interaction of the body with the context (= objects / persons) is also important to correctly classify the action being performed. In this paper, we propose a feature representation scheme for large scale human action recognition in realistic videos.

Amongst several frameworks, local representation based Bag-of-Words (BoW) techniques are very popular and yield good results. Firstly, interest points are detected at different spatio-temporal locations and scales for a given video. Then, local feature descriptors are computed in the spatio-temporal neighbourhood of the detected interest points, which capture shape (gradient) or motion (optical flow) or similar measurements describing the human action dynamics. Several techniques to detect interest points exist. Laptev and Lindeberg [11] first proposed the usage of Harris 3D corners as an extension of the traditional Harris corner points. Later on, cuboid detectors obtained as local maxima of the response function of temporal Gabor filters on a video were proposed by Dollár *et al.* [4]. Willems *et al.* [29] proposed a Hessian interest point detector, which is a spatio-temporal extension of the Hessian saliency measure for blob detection in images. Wang *et al.* [28] proposed a dense sampling approach wherein the interest points are extracted at regular positions and scales in space and time. Spatial and temporal sampling are often done with 50% overlap. Further, Wang *et al.* [26] extend the dense sampling approach by tracking the interest points using a dense optical flow field. We have observed that trajec-

tories obtained by a Kanade-Lucas-Tomasi (KLT) tracker [8], densely sampled [26] or one of the variants [6, 7, 9, 27], have been consistently performing well on several benchmark action recognition datasets. Early work by Albright *et al.* [2] indicates that is it sufficient to distinguish human actions by tracking the body joint positions. Hence, we focus our current work on trajectories only.

We divide the existing trajectory based techniques into three major approaches. In the first approach, a variant of trajectories [7, 9] and/or new local feature descriptors [6] is proposed. In the second approach, feature histograms are computed in different volumes obtained my dividing the video along height, width and time, and then aggregated. This approach is akin to the popular Spatio-Temporal Pyramidal approach [12, 13, 24]. The trajectories and their variant can also be aggregated together. In the third approach, only some trajectories are selected from those obtained by the above first or second approach. All three categories for feature representation are still in practice and found to yield good results on different action recognition datasets. However, their effectiveness on new large scale action recognition datasets, which contain many more classes, and videos collected in realistic conditions has not yet been studied extensively. Our proposed approach falls into the third category. We believe that in large scale datasets, it is important to have a smaller but richer set of features for efficient action recognition. Thus, we propose a scheme to match dense trajectories in consecutive video frames to select only a few trajectories and then to generate a set of ordered trajectories. We study and present our results on large scale action datasets such as **HMDB51**, **UCF50** and **UCF101** containing at least 50 different action classes. All our experiments are performed in a BoW framework using a Support Vector Machine (SVM) classifier. In this paper, we make the following contributions:

1. A feature selection like approach that selects about half of the dense trajectories, yet delivers better performance than the original and several other trajectory variants.

2. Removal of a large number of trajectories related to background noise.

In the remainder of the paper, Section 2 contains a review of the latest advances in feature representations w.r.t. large scale action recognition. Section 3 describes the framework. Section 4 details the local feature descriptors, codebook generation, classifier and datasets. Section 5 presents and discusses the results obtained on the benchmark datasets. Finally, conclusions are drawn in Section 6.

2. Related Literature

We now briefly review some of the literature related to the three kinds of approaches mentioned in Section 1. We focus particularly on trajectory based techniques in the last five years as they have been found to perform better than most other techniques on larger action recognition datasets, e.g. **UCF50** and **HMDB51**, with 50 or more action classes.

2.1. Trajectories and Variants

Uemura *et al.* [23] proposed human action recognition based on the KLT tracker and SIFT descriptor. Multiple interest point detectors were used to provide a large number of interest points for every frame. Sun *et al.* [22] proposed a hierarchical structure to model spatio-temporal contextual information by matching SIFT descriptors between two consecutive frames. Actions were classified based on intra- and inter-trajectory statistics. Messing *et al.* [15] proposed an activity recognition model based on the velocity history of Harris 3D interest points (tracked with a KLT tracker). Matikainen *et al.*[14] proposed a model to capture the spatial and temporal context of trajectories, which were obtained by tracking Harris corner points in a given video using a KLT tracker.

Wang *et al.* [26] proposed dense trajectories to model human actions. Interest points were sampled at uniform intervals in space and time, and tracked based on displacement information from a dense optical flow field. Kliper-Gross *et al.* [9] proposed Motion Interchange Patterns (MIP) for capturing local changes in motion trajectories. Based on dense trajectories [26], Jiang *et al.* [7] proposed a technique to model the object relationships by encoding pairwise dense trajectory codewords. Global and local reference points were adopted to characterise motion information with the aim of being robust to camera movements. Jain *et al.* [6] proposed another variant of dense trajectories recently, showing that significant improvement in action recognition can be achieved by decomposing visual motion into dominant (assumed to be due to camera motion) and residual motions (corresponding to the scene motions).

2.2. Spatio-Temporal Pyramidal Approach

To encode spatial information within the bag-of-features representation, the *Spatial Pyramidal* approach was first proposed by Lazebnik *et al.* [13] for object classification in images, but has since also been successfully used in videos. Here, the video sequence is split into (spatial) subsequences and a histogram is computed for each subsequence. The final histogram is obtained by concatenating or accumulating all the histograms of the subsequences. Zhao *et al.* [30] divide each frame into cells, over which Dollár's features [4] are computed. Additionally, motion features from neighbouring frames are used in a weighted scheme, which takes into account the distance of the neighbour from the actual frame. A spatial-pyramid matching, similar to [13], is then applied to compute the similarity between frames. Finally, frames are classified individually and a voting scheme is

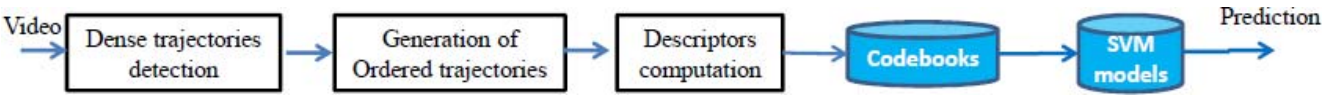

Figure 1: Dense trajectories are extracted from video. Proposed ordered trajectories are generated by matching trajectories of consecutive frames. Codebooks are constructed for each set of local feature descriptors of these trajectories. Separate SVMs are built and the final decision is obtained by *product* fusion rule.

used for action recognition in the video. Ullah *et al.* [24] used six spatial subdivisions of a video and computed local features and models for each subdivision. They found significant improvement in the recognition rate compared to using the original video.

2.3. Interest Point Selection

Nowak *et al.* [16] showed by extensive experiments that uniform random sampling can provide performances comparable to dense sampling of interest points. Another independent study [25] showed that action recognition performance can be maintained with as little as 30% of the densely detected features. Bhaskar *et al.* [1] presented an approach for selecting robust STIP detectors by applying surround suppression combined with local and temporal constraints. They show that such a technique performs significantly better than the original STIP detectors. Shi *et al.* [19] proposed that with proper sampling density, a state-of-the-art performance can be achieved by randomly discarding up to 92% of densely sampled interest points.

3. Overall Framework and Background

The overall layout of our proposed framework is shown in Fig. 1. Firstly, dense trajectories are detected. The proposed ordered trajectories are generated by matching trajectories of consecutive frames. Local descriptors – Motion Bound Histograms (MBH), Histograms of Oriented Gradients (HOG), Histogram of Optical Flow (HOF) – are computed at the selected matching trajectories, while the *Trajectory Shape* descriptor is computed from the generated ordered trajectories. These descriptors are clustered into a pre-defined number of centres, thus constituting codebooks for the rest of the process. Using the cluster centres, the local descriptors extracted from a given video clip are quantised into feature vectors. These frequency histograms act as features to learn a classifier (SVM). Separate codebooks are constructed for each type of descriptor. Decision values from each SVM are fused using the product rule to predict the action in a given video. Our work is primarily inspired by the dense trajectories proposed by Wang *et al.* [26], which we briefly summarise next.

3.1. Dense Trajectories

Firstly, points uniformly spaced over each frame in 8 spatial scales are sampled. This ensures that points are equally spread in all spatial positions and scales. By experimentation, [27] report that a sampling step size of $W = 5$ pixels yields good results over several benchmark datasets. Points in homogeneous regions are removed by applying the criterion of Shi and Tomasi [20]. The sampled points are then tracked by applying median filtering over the dense optical field. For a given frame i_t, its dense optical flow field $\omega_t = (u_t, v_t)$ is computed w.r.t. the next frame i_{t+1}, where u_t and v_t are the horizontal and vertical components of the optical flow. A point $P_t = (x_t, y_t)$ in frame I_t is tracked to another position $P_{t+1} = (x_{t+1}, y_{t+1})$ in frame I_{t+1} as follows

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega_t)|_{(x_t, y_t)} \qquad (1)$$

where M is a 3×3 pixels median filtering kernel.

The algorithm proposed by Farneback [5] was used to compute dense optical flow. To avoid drifting from their initial locations during the tracking process, tracking is performed on a fixed length L number of frames at a time. Through experimentation, $L = 15$ frames was found suitable. In a post-processing stage, trajectories with sudden large displacements are removed.

3.2. Local Feature Descriptors

Four kinds of local feature descriptors were computed on the neighbourhood of the points derived by the above trajectories. They are MBH, HOG, HOF and Trajectory Shape. Each descriptor captures some specific characteristics of the video content. HOG descriptors capture the local appearance, while HOF descriptors capture the changes in the temporal direction. The space-time volumes (spatial size 32×32 pixels) around the trajectories are divided into 12 equal-sized 3D grids (spatially, 2×2 grids and temporally, 3 segments). For computing HOG, gradient orientations were quantised into 8 bins. For computing HOF, 9 bins were used with one more zero bin in comparison to HOG. Thus, the HOG descriptors have 96 dimensions and HOF descriptors have 108 dimensions. MBH descriptors are based on motion boundaries. These descriptors are computed by separate derivatives for the horizontal and vertical components of the optical flow. As MBH captures the gra-

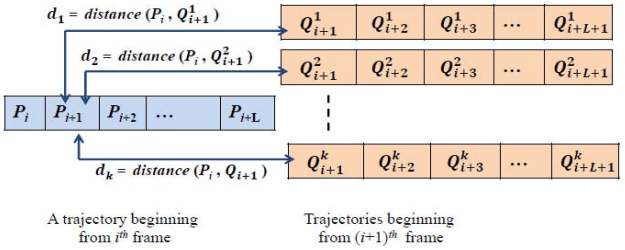

Figure 2: A trajectory of points beginning in frame i searches for matching trajectory of points beginning from frame $i+1$ based on the distance between the second point in the former and the first point in the latter trajectories.

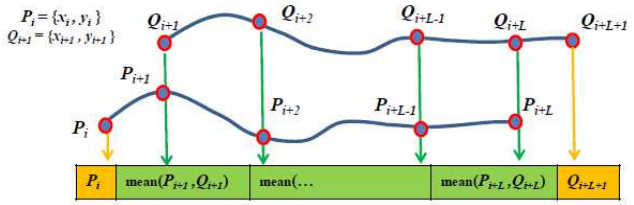

Figure 3: The first point in the trajectory of frame i and the last point $i+L+1$ in the trajectory of frame $i+1$ are transferred as first and last points of the new ordered trajectory, respectively. The remaining trajectory points are merged in an ordered manner by taking the mean as the intermediate elements of the new trajectory.

dient of the optical flow, constant camera motion is removed and information about changes in the flow field (i.e. motion boundaries) are retained. An 8-bin histogram is obtained along each component of x and y. Both histogram vectors are normalised separately with their L_2 norm, each becoming a 96-dimensional vector. In our experiments, we built separate codebooks for MBH descriptors along x and y. The trajectory shape descriptor encodes local motion patterns. For a trajectory of given length L (number of frames) and containing a sequence of points ($P_t = (x_t, y_t)$), the trajectory shape is described in terms of a sequence of displacement vectors $\Delta P_t = (P_{t+1} - p_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. The resulting vector is normalised by the sum of displacement vector magnitudes

$$T = \frac{\Delta P_t, \ldots, \Delta P_{t+L-1}}{\sum_{j=t}^{t+L-1} ||\Delta P_j||} \qquad (2)$$

For $L = 15$ frames, a 30-dimensional trajectory shape descriptor is obtained. The dense trajectories code available online[1] [26] was used in all our experiments.

4. Proposed Ordered Trajectories

In this section, we describe the generation of our proposed ordered trajectories from the dense trajectories obtained earlier. There are two stages in this process: the matching stage and the generation stage.

4.1. Search for a Matching Trajectory

For any dense trajectory, we search for any matching trajectory that potentially exists and begins from immediate next frame. The approach is described now. Consider a trajectory $P_i = (x_i, y_i)$, $i = 1, 2, \ldots, L$ beginning from frame i. Potential matching trajectory is sought among all the trajectories $Q^k_{i+1} = (x^k_{i+1}, y^k_{i+1})$, beginning from frame $i+1$. For a trajectory to match / merge into another trajectory beginning from the next frame, ideally, the second point in the former trajectory should be the closest possible one to the first point of the latter trajectory. This inspiration is used for searching the matching trajectory (see Fig. 2).

We pick up the second point $P_{i+1} = (x_{i+1}, y_{i+1})$ from the former trajectory and compute its Euclidean distance $distance(P_i, Q^k_{i+1})$ to all first points of the latter trajectories (beginning from the immediate next frame). The smallest distance reveals the identity of potential matching trajectory. However, this 'smallest' is relative. To make it absolute, we further impose threshold for a trajectoy to be selected as matching. This threshold is determined as $5\sqrt{2}$, owing to the $W = 5$ pixels uniform spacing chosen in computing the dense trajectories. The matching of two trajectories is, thus, reduced to matching of 2D points, instead of matching along their full length (L). As this computation is done only once, separate for each video, this is computationally inexpensive.

4.2. Generation of an Ordered Trajectory

This stage consists of merging two matched dense trajectories into a new sequence of points, the *ordered trajectory*. Consider two trajectories of points, $P_i = (x_i, y_i)$ and $Q_{i+1} = (x_{i+1}, y_{i+1})$ found to be matching. The first and last elements of the ordered trajectory consist of $P_i = (x_i, y_i)$ and $Q_{i+L+1} = (x_{i+L+1}, y_{i+L+1})$, respectively (see Fig. 3). The intermediate elements of an ordered array are filled by taking the mean of the remaining elements of P_{i+1} and Q_{i+1}. The pseudo-code to generate ordered trajectories for an entire video is given in Alg. 1.

4.3. Bag-of-Words Framework

We built separate dictionaries for each descriptor. In our experiments, dictionaries are constructed with k-means clustering. We set the number of visual words V to 4000, which was shown to give good results [28]. We have tried two types of encoding techniques as follows.

[1] http://lear.inrialpes.fr/software

Algorithm 1: Generating ordered trajectories

Input: Dense trajectories detected in a video
Output: Ordered dense trajectories for the video

$N \leftarrow$number of frames in the video
for $i \leftarrow 1$ ***to*** $N-1$ **do**
  $P_i^J \in$ Trajectories beginning from i^{th} frame
  $Q_{i+1}^K \in$ Trajectories beginning from $i+1^{th}$ frame
  Count1 $\leftarrow 1$
  for $j \leftarrow 1$ ***to*** J **do**
    $dist = 0$
    Count2 $\leftarrow 1$
    for $k \leftarrow 1$ ***to*** K **do**
      $dist[Count2]) = ||P_{i+1}^j - Q_{i+1}^k)||_2$
      Count2 $\leftarrow$ Count2 $+1$
    $min\_distance = min(dist)$
    if $min\_distance < threshold$ **then**
      $PQ_i[Count1] = [p_i^j mean(p_{i+1}^j, Q_{i+1}^k)$ ...
      $...mean(p_{i+L}^j, Q_{i+L}^k) Q_{i+L+1}^k]$
      Count1 $\leftarrow$ Count1 $+ 1$
return *Ordered trajectories PQ*

Hard Assignment (HA): After creating a codebook $C = \{\mu_1, \mu_2, ..., \mu_k\}$, local descriptors are assigned to the closest centroid as

$$q : R^d \rightarrow C \subset R^d \tag{3}$$

$$x \mapsto q(x) = \arg\min_{\mu \in C} ||x - \mu||^2 \tag{4}$$

where the norm operator $||.||_2$ refers to the L_2 norm.

Vector of Locally Aggregated Descriptors (VLAD): This is a very recent technique applied successfully on action recognition by [6]. In this encoding, the difference between the descriptors and the closest centroid is collected as residual vectors. For each centroid of dimension d (dimension of local feature descriptor), a sub-vector v^i is obtained by accumulating these residual vectors as

$$v^i = \sum_{x:q(x)=\mu_i} x - \mu \tag{5}$$

The obtained sub-vectors are concatenated to yield a D-dimensional vector, where $D = k \times d$.

Further, two-stage normalisation is applied. Firstly, the 'power-law normalisation' [17] is applied. It is a component-wise non-linear operation. Each component $v_j, j = 1$ to D is modified as $v_j = |v_j|^\alpha \times sign(v_j)$, where α is a parameter such that $\alpha \leq 1$. In all our experiments, $\alpha = 0.2$. Finally, the vector is L_2-normalised as $v = \frac{v}{||v||}$ to yield the VLAD vector. When VLAD encoding is used, we use a linear SVM for classification.

4.4. Classification

For HA encoded features, we build a non-linear SVM with an RBF-χ^2 kernel classifier for each descriptor separately. We compute the χ^2 distance for each pair of training feature vectors and obtain the kernel matrix. We normalise this kernel matrix using the average χ^2 distance value A of the training samples within themselves. We then map this Kernel matrix using exponential function e^{-x}, i.e.

$$K(H_i, H_j) = exp(-\frac{1}{2A}\frac{(H_i - H_j)^2}{H_i + H_j}) \tag{6}$$

$H_i = h_{in}$ and $H_j = h_{jn}$ are the frequency histograms of word occurrences.

We next feed this transformed kernel matrix to an SVM with a Radial Basis Kernel [3]. We have fixed the hyper parameter $C = 100$ in all our experiments. For multi-class classification, we apply the one-versus-all approach and select the class with the highest score.

4.5. Datasets

We applied our proposed technique on three benchmark datasets: **HMDB51** [10], **UCF50** [18] and **UCF101** [21]. **HMDB51** contains 51 actions categories. Digitised movies, public databases such as the Prelinger archive, videos from YouTube and Google videos were used to create this dataset. For evaluation purposes, three distinct training and testing splits were specified in the dataset. These splits were built to ensure that clips from the same video were not used for both training and testing. For each action category in a split, 70 training and 30 testing clips indices were fixed so that they fulfil the 70/30 balance for each meta tag. **UCF50** has 50 action classes. This dataset consists of 6680 realistic videos collected from YouTube. As specified by [18], we evaluate 25-fold group wise cross-validation classification performance on this dataset. **UCF101** data set is an extension of **UCF50** with 101 action categories, also collected from realistic action videos, e.g. from YouTube.

5. Results and Discussions

We now present the results obtained by applying the proposed ordered trajectories. These results are obtained by applying HA encoding. Results obtained by VLAD encoding are presented only in Section 5.5.1.

5.1. Performance of Ordered Trajectories

Results obtained by using the ordered trajectories on **HMDB51**, **UCF50** and **UCF101** dataset are presented in Table 1. The overall performance on **HMDB51** and **UCF50** datasets is 47.3% and 85.5%, which is 0.7-1% (absolute) more than the traditional dense trajectories [26] computed at 5 different scales in spatial and time [27]. The performance of individual local feature descriptors shows that

Table 1: Ordered trajectory based recognition rates

Approach	Descriptor	UCF50	HMDB51	UCF101
Dense Trajectories [26]	TrajShape	67.2%	28.0%	NA
	MBH	82.2%	43.2%	NA
	HOG	68.0%	27.9%	NA
	HOF	68.2%	31.5%	NA
	Combined	84.5%	46.6%	NA
Proposed approach	TrajShape	72.4%	31.2%	47.1%
	MBH	82.3%	41.0%	67.9%
	HOG	66.2%	26.5%	51.4%
	HOF	67.0%	30.9%	52.0%
	Combined	**85.5%**	**47.3%**	**72.8%**

only the trajectory shape descriptor performs better than the Spatio-Temporal Pyramidal approach by 3-4% (absolute) due to the trajectory descriptor being computed on the merged matching dense trajectories. In the case of other descriptors – MBH, HOG, HOF – we only retain that of matching trajectories.

5.2. Minimising Background Clutter

The effect of the proposed ordered trajectories can be observed in Fig. 4. The ordered trajectories have not lost the principal object, players or the basketball. At the same time, some background clutter, e.g. roof top, has been filtered out by the proposed technique.

5.3. Ordered Trajectories on a Sample Video

For a sample video of 13s duration, 819KB file size, 320×240 pixels frame size, the number of dense trajectories obtained, by original and different variants is shown in Table 2. Dense trajectories computed at original scale in space and time, as shown in Fig. 5 (a) is 21,647. By the proposed technique, 11,657 ordered trajectories were obtained, which is about 50% of the actual dense trajectories amount. A natural question that may follow is *Why not compute dense trajectories directly for* $L = 16$*?*. Actually, dense trajectories are first detected at different scales in space. A trajectory that may have ended in one scale, may have continued in the next frame in the upper or lower scale. Dense trajectories do not exclusively combine such trajectories. The number of dense trajectories computed for $L = 16$ and $L = 17$ are shown in Table 2. Although their number is slightly smaller than for $L = 15$, it is still nearly twice the number of our proposed ordered trajectories.

5.4. Comparison with Spatio-Temporal Pyramid

In the *Spatio-Temporal Pyramidal* approach, features are computed separately in each of the different sub-volumes (obtained by dividing the original video along space and time) and then concatenated. Illustrations are given in Fig. 5. For example, when the original video is divided into three

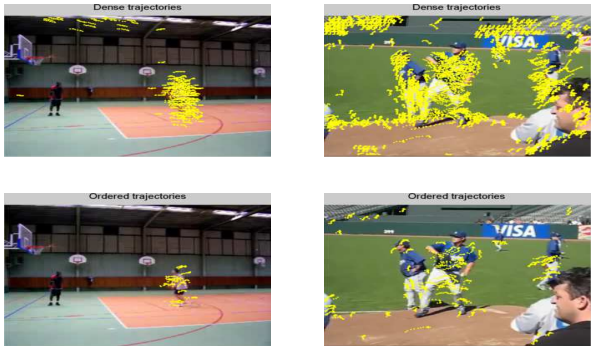

Figure 4: *Top:* Original dense trajectories in sample frames. *Bottom:* Ordered dense trajectories in corresponding frames. Watch full videos in supplementary material.

horizontal sub-volumes, as shown in Fig. 5(b) three separate codebooks are to be constructed for each sub-volume. Using those dense trajectories that fall in a particular sub-volume, feature histograms are generated. These feature histograms are concatenated to yield the overall feature vector for the video. In current case, the feature vector will be 3 times larger than that obtained on full scale dense trajectories or our proposed ordered trajectories. In a higher case, as can be seen in Fig. 5(f) where the original video is divided into eight sub-volumes, the final vector will be 8 times larger than that that can be obtained by original dense or ordered trajectories.

5.5. Comparison with Other Techniques

For a fair and consistent comparison, we compared our results with the latest, best and relevant works in the literature. We also include a brief description of their methodology in Table 3. The performance of dense trajectories [26] on **HMDB51** and **UCF50** is 84.5% and 46.6%, respectively,

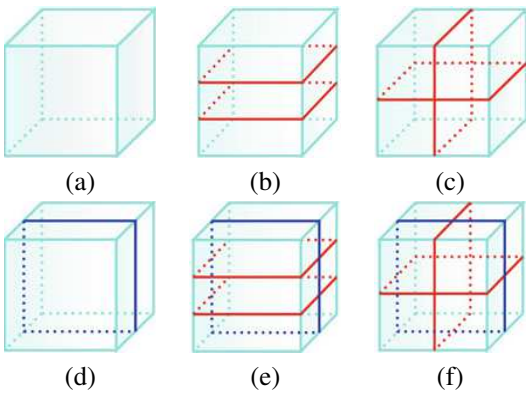

Figure 5: Spatio-temporal grids used by Wang *et al.* [27]

Table 2: Number of trajectories for a sample video in different schemes.

Reference	Number of Trajectories
Dense trajectories [26] at $L = 15$	21,647
Dense trajectories at $L = 16$	21,521
Dense trajectories at $L = 17$	21,329
Proposed approach	**11,657**

as reported in [27]. The performance by using the proposed ordered trajectories on respective datasets is 85.5% and 47.3% respectively. This is nearly 0.7-1% (absolute) better than for the actual dense trajectories. The results obtained by the *Spatio-Temporal Pyramidal* approach of dense trajectories is 85.6% and 48.3% respectively. This is 0.1-1% better than our proposed technique. However, it has to be noted that our results are obtained using only 5 channels while the *Spatio-Temporal Pyramidal* approach uses 30 channels. The feature vectors in some of these channels can also be as high as 3-8 times than those used in original dense trajectories or our proposed ordered trajectories. Shi *et al.* 2013 [19] suggested that random selection of 10,000 dense trajectories can yield 83.3% and 47.6%, respectively. By our proposed technique, we see that only less important information such as background is omitted, as shown visually in a few examples in Fig. 4.

5.5.1 VLAD Based Results

The latest work on the **HMDB51** dataset by Jain *et al.* reported the highest recognition rate so far of 52%. This can be due to two reasons. Firstly, due to the combination of dense trajectory results and ω-trajectories (motion compensated) results. Secondly, due to the new encoding scheme VLAD. We have used the same encoding scheme on our ordered trajectory descriptors and include our results. Dense trajectories with VLAD encoding yielded 48.0% [6], while ordered trajectories with VLAD encoding yielded 49.9%. It can be observed that our results are 1.9% (absolute) better even when VLAD encoding is used.

5.6. Future Work

After performing the experiments and the selection of trajectories, we would like to focus on the choice of local feature descriptors. There are several types of local feature descriptors already available in the literature. We would like to investigate if with minimal choice we can obtain the state-of-the art performance.

6. Conclusions

A technique to match dense trajectories and merge them is proposed. The resultant sequence of points were coined as *ordered dense trajectory*. A local descriptor, trajectory shape, computed over the ordered dense trajectories was found to yield better performance than that obtained by the original dense trajectories. Moreover, information other than the main objects of interest captured by the dense trajectories was also found out to be removed effectively by the proposed technique. Future work will involve developing tools to merge other descriptors such as HOG, HOF and MBH of the matching dense trajectories.

References

[1] Selective spatio-temporal interest points. *Computer Vision and Image Understanding*, 116(3):396–410, 2012. 4323

[2] T. D. Albright and G. R. Stoner. Visual motion perception. *Proceedings of the National Academy of Sciences*, 92(7):2433–2440, 1995. 4322

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 4325

[4] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behaviour recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. IEEE, Oct. 2005. 4321, 4322, 4328

[5] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conference on Image Analysis (SCIA'03)*, pages 363–370. Springer-Verlag, 2003. 4323

[6] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Apr. 2013. 4322, 4325, 4327, 4328

[7] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In *European Conference on Computer Vision (ECCV 2012)*, pages 425–438, 2012. 4322, 4328

[8] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG2000)*, pages 46–53, 2000. 4322

[9] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion Interchange Patterns for Action Recognition in Unconstrained Videos. In *European Conference on Computer Vision (ECCV 2012)*, pages 256–269, Oct 2012. 4322, 4328

[10] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *International Conference on Computer Vision (ICCV)*, pages 2556–2563, nov 2011. 4325, 4328

[11] I. Laptev and T. Lindeberg. Space-Time Interest Points. In *International Conference on Computer Vision (ICCV)*, pages 432–439, Oct 2003. 4321

[12] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2008. 4322

Table 3: Comparison with other techniques

Approach	Brief description	UCF50	HMDB51	UCF101
Kuehne *et al.* 2011 [10]	C2 features inspired by visual cortex	47.9%	22.8%	NA
Kliper-Gross *et al.* 2012 [9]	Motion Interchange patterns	72.6%	29.2%	NA
Reddy and Shah 2012 [18]	Cubiod detectors [4]	76.9%	NA	NA
Khurram *et al.* 2012 [21]	Harris 3D corners	NA	NA	44.5%
Wang *et al.* 2011 [26]	Dense trajectories	84.5%	46.6%	NA
Jiang *et al.* 2012 [7]	Dense trajectory variant	NA	39.8%	NA
	proposed dense trajectory variant + dense trajectories		40.7%	
Wang *et al.* 2013 [27]	Dense trajectories in Spatio-Temporal Pyramid	85.6%	48.3%	NA
Feng Shi *et al.* 2013 [19]	Random sampled local spatio-temporal features	83.3%	47.6%	NA
Jain *et al.* 2013 [6]	Dense trajectories + VLAD encoding	NA	48.0%	NA
	Dense trajectories variant + dense trajectories + VLAD encoding		**52.1%**(See 5.5)	
Proposed approach	Ordered trajectories(+ HA encoding)	85.5%	47.3%	72.8%
	Ordered trajectories + VLAD encoding	**87.3%**	49.9%	**73.1%**

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pages 2169–2178, 2006. 4322

[14] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *European conference on Computer vision (ECCV 2010)*, ECCV'10, pages 508–521. Springer-Verlag, 2010. 4322

[15] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *International Conference on Computer Vision (ICCV 2009)*, pages 104–111, 2009. 4322

[16] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision (ECCV 2006)*, volume 3954 of *Lecture Notes in Computer Science*, pages 490–503. Springer, 2006. 4323

[17] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on Computer vision (ECCV 2010)*, pages 143–156. Springer-Verlag, 2010. 4325

[18] K. K. Reddy and M. Shah. Recognizing 50 Human Action Categories of Web Videos. *Machine Vision and Applications*, 24:971–981, Sept 2012. 4325, 4328

[19] F. Shi, E. Petriu, and R. Laganière. Sampling strategies for real-time action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, Apr. 2013. 4323, 4327, 4328

[20] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Vision and Pattern Recognition (CVPR 1994)*, pages 593–600, 1994. 4323

[21] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A Dataset of 101 Human Action Classes from Videos in the Wild. In *CRCV-TR-12-01*. November 2012. 4325, 4328

[22] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 2004–2011, 2009. 4322

[23] H. Uemura, S. Ishikawa, and K. Mikolajczyk. Feature tracking and motion compensation for action recognition. In *British Machine Vision Conference (BMVC 2008)*, pages 30.1–30.10, 2008. 4322

[24] M. M. Ullah, S. N. Parizi, and I. Laptev. Improving Bag-of-Features Action Recognition with Non-Local Cues. In *British Machine Vision Conference (BMVC 2010)*, pages 95.1–95.11. BMVA Press, 2010. 4322, 4323

[25] E. Vig, M. Dorr, and D. Cox. Space-Variant Descriptor Sampling for Action Recognition Based on Saliency and Eye Movements. In *European Conference on Computer Vision (ECCV 2012)*, volume 7578 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2012. 4323

[26] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 3169–3176, June 2011. 4321, 4322, 4323, 4324, 4325, 4326, 4327, 4328

[27] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013. 4322, 4323, 4325, 4326, 4327, 4328

[28] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference (BMVC 2009)*, pages 124.1–124.11, Sep 2009. 4321, 4324

[29] G. Willems, T. Tuytelaars, and L. Gool. An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. In *European Conference on Computer Vision (ECCV 2008)*, pages 650–663, 2008. 4321

[30] Z. Zhao and A. Elgammal. Human activity recognition from frameś spatiotemporal representation. In *International Conference on Pattern Recognition (ICPR 2008)*, pages 1–4, Dec. 2008. 4322