Query-based Retrieval of Complex Activities using "Strings of Motion-Words"

Utkarsh Gaur, Bi Song, Amit K. Roy-Chowdhury *
University of California, Riverside
{ugaur,bsong,amitrc}@ee.ucr.edu

Abstract

Analysis of activities in low-resolution videos or far fields is a research challenge which has not received much attention. In this application scenario, it is often the case that the motion of the objects in the scene is the only low-level information available, other features like shape or color being unreliable. Also, typical videos consist of interactions of multiple objects which pose a major vision challenge. This paper proposes a method to classify activities of multiple interacting objects in low-resolution video by modeling them through a set of novel discriminative features which rely only on the object tracks. The noisy tracks of multiple objects are transformed into a feature space that encapsulates the individual characteristics of the tracks, as well as their interactions. Based on this feature vector, we propose an energy minimization approach to optimally divide the object tracks and their relative distances into meaningful partitions, called "strings of motion-words". Distances between activities can now be computed by comparing two strings. Complex activities can be broken up into strings and comparisons done separately for each object or for their interactions. We test the efficacy of our approach to search all the instances of a given query in multiple real-life video datasets.

1. Introduction

Activity recognition is one of the most interesting and complex problems in computer vision. Most activity recognition work has concentrated on analysis of simple activities (running, waving, jumping) in relatively high resolution video, by which we mean videos where the shape and appearance of the objects provides meaningful discriminating information. This is evidenced in standard datasets like KTH, Wiezman or IXMAS [12, 5, 19] and most of the well-known algorithms. A challenging domain of activity recognition that has received lesser attention is when the actions are in a far-field and the objects are of a low-resolution. In these cases, the appearance information is exceedingly unreliable and all that we have in terms of low-level features is a noisy track of each object. Also, typical videos in this domain consist of multiple interacting objects whose activities need to be modeled and recognized

Figure 1: Some examples of low-resolution videos with some parts magnified.

In this paper, we look at activities in low-resolution video, like the examples in Fig. 1. The goal is to work with the noisy tracks of the objects and recognize activities that are defined solely by the tracks. Examples include motions of cars and people on a ground plane as observed from the top of tall building or from the air. Many interesting activities in this scenario involve interactions between objects, e.g., people entering/exiting buildings, cars moving along specific paths, groups of people meeting, and so on. Our proposed method not only identifies the actions of each object separately, but is also capable of modeling and recognizing the interactions between multiple objects.

1.1. Brief Overview of Proposed Method

The core of the proposed method involves a transformation of the *noisy* tracks of the multiple objects into a motion feature space that encapsulates the individual characteristics of the tracks, as well as their interactions. Each individual track is represented by its gradients as a function of time. Each pair of tracks is represented by the relative distance between the components as a function of time. By considering multiple pairs, interactions between more than two objects can be modeled in an iterative manner. Thus each pair of tracks is now represented by a multi-dimensional feature vector that, at each time instant, consists of the gradients of each track and the distance between the tracks. We shall call this the GRD (gradient + relative distance) feature vector. These transformed motion features, which are a function of time, have the ability to capture the global characteristics of the motion of multiple objects.

* The authors were supported by ARO grant W911NF-07-1-0485, and DARPA VIRAT program.

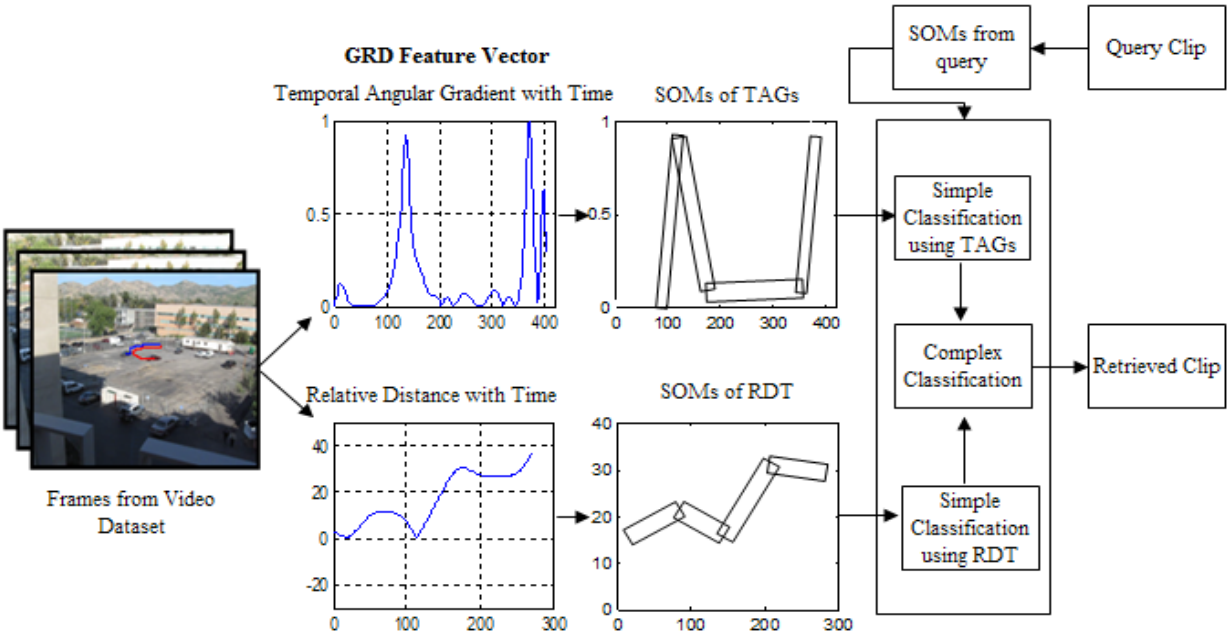

Figure 2: A diagrammatic representation of the activity modeling and query classification approach. The "string of motion-words" is shown in black using double lines.

Instances of a same activity could vary in realization due to the semantic nature of the definition of an activity. To come up with a common semantic grouping of activities based on the observed statistics calls for features which lead to least intra class variance. To this end, by analyzing the relative change in directions and distance, rather than the absolute position, invariance to affine transformations in the activity sequences being match is achieved.

An additional challenge that we have to deal with is robustness to the tracking errors, which are bound to be present in low resolution video. Towards this goal, the feature vs. time plots are broken down into segments so that each segment constitutes a particular characteristic of the tracks. This will be achieved by clustering on gradient and relative distance in the GRD feature space. For example, the track of a car coming straight, taking a right turn and then again proceeding straight would ideally have three segments. This allows us to compensate for noise and ignore outliers in the tracking method. We will show how to automatically obtain these segments through an energy-minimization approach. Analogous to the bag-of-video-words approaches [7, 11], we can think of each cluster as a "word" in the feature space. However, the clusters (segments) have a strict time ordering since that is critical for motion analysis (this is unlike the bag of video-words approaches). Hence each feature time-series is now represented as a "**string of motion-words**" (SOMs), where "string" implies that the motion-words are tied together with a particular temporal ordering. This "string" representation is robust to scaling and rotation of the tracks on the ground plane, which is usually sufficient for far-field activities.

Similarity between two activities can be computed by comparing their "strings". This would involve computing the similarity in the gradient space for each individual track, and the similarity of each pair in the relative distance space. To account for speed and scale variations, this distance computation will be done using Dynamic Time Warping (DTW) on the strings. The modeling and classification approaches are shown in Fig. 2. The proposed method will be evaluated for querying in video databases given a single query clip. The transformed motion features and string representations will allow us to compare different kinds of activities. For example, if there are two cars following each other in the query, we can retrieve *separately* examples from the database of cars following irrespective of their trajectory, as well as, examples where the following happens along a particular trajectory.

1.2. Related Work

A good amount of research has been done on activity recognition but most of it assumes high resolution data and cannot be extended to lower resolutions. A survey of some of the work on activity modeling and recognition clearly shows that most of the methods are tuned for high resolution data [5, 6, 7, 8, 9, 10, 13]. This is also apparent by analyzing some of the commonly used activity recognition datasets (e.g., Wiezman [5], KTH [12] and IXMAS [19]). The authors in [16] proposed a method for recognizing low-resolution activities by modeling the shape of the tracks of the objects. However, generalization of the approach is difficult since it relies on learning dynamical models to describe the interactions. Learning such models as the number of objects increases is impractical. The transformed motion features in our case will not require the identification of such models a-priori,

while still allowing us to identify activities of individual objects and their interactions. The research in [3, 20] analyses low-resolution videos but requires periodicity be present in the motion.

Availability of large training sets has been another assumption for developing activity recognition systems. Liu et al [7] propose learning of an optimized codebook for human action analysis. The research work in [4] extracts sparse spatio-temporal features which are used to perform matching across behaviors in video. The authors of [21] learn semantic visual vocabularies of actions by motion feature pruning based on spatial and temporal feature statistics. The use of multiple features for human action recognition is proposed in the work in [8] where seemingly heterogeneous features are embedded in a common graph. All these approaches assume existence of a large number of examples which is impractical for recognizing complex interactions, given the large number of possibilities. We look at how to retrieve activity clips given a single query from the user.

There has been some work on activity recognition based on tracks obtained from the video under consideration Parameswaran et al [14] extend the cross-ratios for trajectories [15] in two and three dimensional spaces and successfully apply in the domain of human action recognition. This approach does not model the interactive/complex interactions as required in our problem domain. Rao el al [15] learn the spatio-temporal curvature based view-invariant features from the tracks of the hand of actors performing some action out of a predefined list of actions. Their approach also does not model complex/interactive activities between multiple objects. Ali et al [1] learn chaotic invariants from the tracks of body joints which are used to recognize human actions. In addition to the assumption of the tracks of each body joint, it is not easy to generalize their approach to model complex/interactive activities.

1.3. Contributions

The work presented in this paper advances the state-of-the-art by describing features which distinctly model far field activities at a fine level and are loosely coupled with the classification scheme. The latter ensures that the features can be used with a variety of classifiers rather than having to require specific analysis. Our system is robust to noisy tracks obtained by an automatic tracker and to the intra-class activity variations. We model the complex/interactive activities occurring in the scene in addition to the singleton activities to come up with a high level description of the scene. The introduction of "string of motion-words" allows us to obtain the GRD features from noisy tracks.

2. Problem Formulation

Consider that we have two video sequences. One of them, which we will call the video database (D), has tracks (possibly noisy) of P objects over its entire time period T_D. The other sequence, which we call the video query (Q), has tracks of L objects, where $L << P$ and the total time of the query sequence, $T_Q << T_D$. The methodology can be easily generalized to multiple query sequences, so for ease of explanation, we will consider only one sequence. Our problem is to find the parts in D that match Q. Note that the match may be only over a partial track in D.

This is a situation where we have to look for certain kinds of activities in a database, with some examples being provided by the query. The query would typically have only a few objects (e.g., cars following, people entering a building, two people meeting, and so on), while the database would be much longer and consist of many objects and their activities. For example, a query could be of a car taking a particular turn, while the complete track of the car may consist of many other motions.

Our proposed solution to this problem has the following parts:

(i) Extraction of low-level features (GRD) from the tracks of Q and D. The features will be specific to a single track (gradients), as well as encode the relationships between pairs of tracks (relative distances).

(ii) In this feature space, we will cluster the features temporally, where each cluster encodes some characteristic of the motion of the objects. These clusters can be thought of as "motion-words", and the entire track as a "string of motion-words". Note that the temporal ordering of these words is critical and the clustering allows us to deal with the noisy data.

(iii) Then, we will match the "strings" from Q with those from D using dynamic time warping on sliding windows on the tracks in D.

3. Motion Feature Extraction

We consider the motion features for both the individual tracks and their interactions. Each individual track is represented by the relative angular orientation of its instantaneous gradients as a function of time. Each pair of tracks is represented by the relative distance between the components as a function of time.

3.1. Temporal Angular Gradients (TAGs) for Single Tracks

The tracks obtained are often noisy resulting from the poor quality of the video. Thus, a locally weighted moving average filter is applied to the noisy tracks to smooth out of the local outliers while maintaining the global motion pattern. This makes our system robust to noisy and broken tracks extracted automatically from the videos.

The direction patterns in the motion of an object could be utilized to uniquely identify its activity. It is observed that the relative angular change of the instantaneous directions of an object is almost similar across different instances of the same activity and additionally it is invariant to affine transformations on the object plane (for far-field activities).

The temporal angular gradients (TAGs) of directions from a trajectory of object j, ϕ_j, are given by

$$G_j = \left\{ \sqrt{2 - 2 * \cos(d(t+1) - d(t))} \mid t = 1 \cdots T - 1 \right\} \quad (1)$$

where, $d(t) = \arctan(y_t, x_t)$ is the instantaneous direction and G is the standard polar distance formula. Thus, TAG is the angular difference of the instantaneous directions using the polar distance formula.

The TAGs are utilized to capture the singleton activity information in the motion pattern of an object's track.

3.2. Relative Distance Estimation

The patterns in the relative distances with time plots are used to identify and classify interactive activities. The relative distance between tracks of two objects is calculated as the Euclidean distance at every time instance.

The reason behind the uniqueness of this feature is fairly intuitive. For instance, in case of a person entering a building, the relative distance plot would decrease and eventually drop to zero. This pattern would be repeated in all instances of the "Entering" interactive activity regardless of the absolute location of the building and the approach direction of the person; that is to say, the pattern in the RD space is invariant to the intra-class variance due to the abstraction from the absolute positions of the person or the building.

4. Strings of Motion-Words

To represent activities with different characteristics in its different parts, we propose a "string of motion-words" representation. Consider the GRD vs. time plot for two interacting objects. This would constitute the TAG and relative distance vs. time plots, each of which can be broken down into segments having some similarity. The motion features of each segment can be quantized and represented as a code-word. We can think of this code-word as a "motion word". Therefore, each motion feature time-series is now represented as a "string of motion-words", where motion-words are tied together in a particular temporal ordering.

4.1. Segmenting motion feature time-series

The problem of segmenting motion feature time-series is equivalent to finding the optimal clusters of units such that the feature coherence of units falling in the same cluster is the maximum. Also, our cluster must obey some temporal constraint, i.e., the units falling in the same cluster must be continuous in time.

This is achieved in a greedy bottom-up maximization approach by merging temporally adjacent pairs of partitions. The algorithm starts with pre-divided small units. We use $W = \{w_1, w_2 \ldots w_n\}$ to denote the series of partitions, where the subscripts indicate the temporal order. The cumulative affinity of the neighboring partitions of the motion feature time-series is defined as:

$$E_W = \frac{1}{n-1} \sum_{i=1}^{n-1} d_{DTW}(w_i, w_{i+1}), \quad (2)$$

where d_{DTW} is the distance between two partitions using Dynamic Time Warping (DTW) algorithm, which allows for some variation in speed.

Let w_j and w_{j+1} be the two candidate adjacent partitions to be merged, and they are merged into $\hat{w}_j$, with the energy after merging being $E_{\hat{W}}$. At each step, we greedily merge the two adjacent partitions provided that $E_{\hat{W}}$ is below a certain threshold. The steps of the algorithms are summarized as follows:

1. Divide the motion feature time-series into small, uniformly spaced partitions.
2. At each step, compute $E_{\hat{W}}$ for all possible merging of adjacent pairs of partitions.
3. Keep merging adjacent pairs until $E_{\hat{W}}$ becomes larger than a predefined threshold E. The details of threshold selection are described in Section 6.

Intuitively, the clustering is done based on the feature affinity of the temporally neighboring segments. The initial small partitions are combined into bigger ones until the minimum feature affinity between the partitions is high. As the algorithm proceeds, similar partitions combine to form a meaningful cluster.

5. Classification by String Comparison

Once each motion feature time-series can be represented by a "string of motion-words" (SOM), the similarity between the query video and the dataset can be computed by comparing their "strings". This would involve computing the similarity in the Temporal Angular Gradient (TAG) space for each individual track, and the similarity of the each pair in the Relative Distance (RD) space.

We consider both the cases of simple query and complex query. If only one individual track or interaction between a pair of tracks is specified in the query video, we call it as a simple query. An example is to find a car making a right turn or two cars maintaining distance without bothering about their individual trajectories. If the query involves the motion of multiple individual objects and their interactions, or there are more than one query videos, we call the case as

complex query. In the above example, if we were concerned about cars following in a particular trajectory, that would be a complex query. In terms of our motion features, simple queries can be represented by *either* of the TAG or RD features, while complex queries will require their combination using the GRD (TAG, RD) features.

The SOM is treated as a combination of two strings, one for the TAG and other for the RD plot. We first explain how to compute the distance between two strings for either the TAG or RD cases (i.e., for simple queries). We then explain how to compute distance for complex case that involves both TAGs and RDs.

5.1. Simple Query

Here we consider the case that there is only one interesting track (either in TAG space for individual object trajectory or in RD space for interaction between a pair of objects) specified by the user in the query video. We denote the SOM of the specified track as S_Q. In the dataset, we are only interested in the tracks which lie in the same space of specified query track, e.g., if the query track is the trajectory of an individual object which lies in TAG space, then we only look at the individual trajectories in dataset, and similarly for interaction between objects. We denote the SOMs of these tracks in dataset as S_D^i, where $i = 1,2...P$, where P is the total numbers of SOMs in dataset for a specific type (TAG or RD).

To match the "string" of the query video with that of the dataset, we use hypothesis testing based on two-class nearest neighbor classification.

We define the length of the string, *Ls*, to be the number of "motion-words" in the string. We assume $Ls(S_Q) = m$, and $Ls(S_D^i) = n, m << n$. We divide S_D^i into overlapping substrings with length m, i.e., the first substring is composed of 1^{st} to m^{th} word, the second substring is composed of 2^{nd} to $(m+1)^{th}$ word and so on. The distance of S_Q with each substring S_{Dsub}^i is defined as:

$$Dis(S_Q, S_{Dsub}^i) = \sum_{j=1}^{m} d_{DTW}(w_Q^j, w_{Dsub}^j) \qquad (3)$$

where d_{DTW} is the DTW distance, and w_Q^j, w_{Dsub}^j are the j^{th} words in S_Q and S_{Dsub}^i respectively. The distance computation is done using Dynamic Time Warping (DTW) on the words to account for speed variations. If $Dis(S_Q, S_{Dsub}^i)$ is less than a predefined decision threshold τ_D then this substring will be classified into the class "similar to query", otherwise, it will be classified into "dissimilar to query".

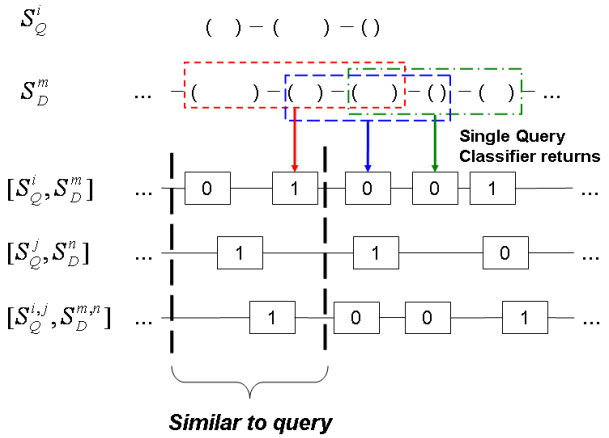

Figure 3: Illustration of proposed complex query algorithm. Three sliding windows on the database strings are shown in different colors. The length of these windows is the same as the length of the query string. 1 indicates that the single-query classifier returns a result that the string in the database is similar to the query string, and 0 indicates they are dissimilar. Once the single-query results on all the strings from the TAG and RD spaces are obtained, we get chains of 1 and 0 as shown above. The maximum time period for a matching link in the chain (i.e., a 1 in the chain) is now considered as a time window. In this time window, if we get at least one "similar" single-query output on each chain, then the portion of the pair of tracks in this window is classified into class "similar to query".

5.2. Complex Query

The action characteristic of an object may consist of both its own moving trajectory and its interaction with other objects. Therefore, we specify the complex query problem as searching for pairs of tracks in dataset which match pairs of tracks giving by the query video. When we say two pairs of tracks match, it means than both the individual trajectories and their interactions match. By exhaustively comparing all possible pairs, we can understand the interactions of more than two objects.

Consider a query video (Q) has tracks of L objects and the tracks are denoted as $R_Q^1, R_Q^2, \cdots R_Q^L$. We could obtain N_Q pairs of tracks where there is an interaction existing within each pair. We take the order of objects in a pair into account, i.e. pair $A_Q^{i,j} = (R_Q^i, R_Q^j)$ and pair $A_Q^{j,i} = (R_Q^j, R_Q^i)$ are different; therefore $N_Q \leq {}^L P_2$. Similarly, we could obtain N_D pairs of tracks from the dataset, which has tracks of P objects. Different from what is done for query video, for each pair of tracks in the dataset, we do not differentiate the pairs according to the order of its elements; that is to say, pair $A_D^{i,j} = (R_D^i, R_D^j)$ and pair $A_D^{j,i} = (R_D^j, R_D^i)$ are the same, so we have $N_D \leq {}^P C_2$.

Giving a pair $A^{i,j} = (R^i, R^j)$, we could obtain three SOMs from it, two for the two individual tracks in TAG space, i.e.,

$S^i = SOM(R^i)$ and $S^j = SOM(R^j)$, and one for their interaction in RD space, i.e., $S^{i,j} = SOM(R^i, R^j)$. Now the problem of comparing $A_D^{m,n}$ with $A_Q^{i,j}$ becomes comparing $[S_D^m \ S_D^n \ S_D^{m,n}]$ with $[S_Q^i \ S_Q^j \ S_Q^{i,j}]$. By comparing S_D^m with S_Q^i, S_D^n with S_Q^j, and $S_D^{m,n}$ with $S_Q^{i,j}$ respectively using the method described in Section 5.1, we will get three chains of single-query classifier returns. 1 indicates that the string in the database is similar to the query string, and 0 indicates they are dissimilar. Once the single-query results on all the strings from the TAG and RD spaces are obtained, we get chains of 1 and 0. The maximum time period for a matching link in the chain (i.e., a 1 in the chain) is now considered as a time window. In this time window, if we get at least one "similar" single-query output on each chain, then the portion of the pair of tracks in this window is classified into class "similar to query". The process is illustrated in Fig. 3.

Scalability: The system is also scalable to handle a complex query constituting of more than two agents. To achieve this, similarity among pair-wise SOMs are calculated for all the possible pairing of objects in the motion space across the video. A video segment is identified as matching the query when the SOMs in the query match the SOMs in the segment for all the pairs. Note that we assume a user will define what he/she wants to be matched in the query.

This framework advances the bag of video-words model by imposing a temporal ordering for the codewords of each activity and having the acuity to model activities involving multiple entities across the video space. This SOM framework is easily generalized to high resolution videos by incorporating other features with the GRD feature vector. This is ongoing work.

6. Experimental Results

Database: To the best of author's knowledge, there is no publicly available dataset for activity recognition in low resolution. In order to test our system, we compiled a database encompassing a broad spectrum of challenging low-level activities. This dataset comprises of three real world scenarios i.e. recordings of a vast construction site, the aerial videos used for DARPA's Video Verification of Identities project (VIVID) [18] and a number of small clips of various atomic interactions between humans and cars in a parking lot. Excluding VIVID, the data was extracted from YouTube videos recorded by amateurs in a relatively unconstrained environment. In all the data, the camera is far from the ground plane where the activities were occurring.

For the construction site dataset, the activities spanned a period of nine minutes. Refer to Fig. 4 (a) for the representative frames of the construction site database. Some of the interesting activities at the construction site include (but are not limited to) people entering a building while walking together, vehicles turning together and stopping, etc. We found this resulting data to be highly challenging in part due to the elevated complexity, induced as a result of complex coupling between various activities and the shear large number of simultaneous activities going on, all happening in a totally unconstrained environment.

The DARPA's VIVID dataset was specifically developed for low-resolution moving target detection, tracking and activity analysis. For our paper, we work with approximately 10 minutes of dataset video which specifically deals with the range of activities targeted in this research.

We also included in our experiments YouTube-extracted short time-period clips of approximately one minute length which capture atomic low-level activities between different vehicles and humans in a parking lot. This dataset comprises of a number of complex and atomic activities usually encountered at a parking lot for instance, people roaming (searching for their vehicle), entering vehicle, vehicles roaming around in the lot, making turns etc. The relatively short length of the clips of this dataset deems it ideal to be used as a query sequence to gauge the performance of retrieval framework. Fig. 4 (b) displays representative frames of this dataset. Thus our total database was for about 20 minutes.

Figure 4: Representative frames for (a) Construction Site Dataset (b) Parking Lot Dataset.

Similarity Computation: Firstly, we demonstrate the uniqueness of the proposed features for efficient classification in form of the similarity matrix for various interactive activities (Fig. 5). Here the numeric value between two activities represents the average of distances of all instances of the two particular activities in the dataset, a smaller number signifying higher similarity. The distance metric used is the DTW distance between the strings of motion words for objects performing the particular activities. It is observed that the two groups of activities: (A1, A3, A7) and (A2, A4, A8) are nearer to each other than to the rest of the activities. This can be explained by the fact that the activity "Entering" is dependent only on the relative distance signature of the participating objects and not on their object class. Moreover, the trajectory for the case of Vehicle stopping/starting is similar to Exiting/Entering. This results in the relative distance signature of "Vehicle Stop/Start" activities ending up close to "Entering/Exiting" respectively. Similarly, the relative

distance signatures of activities "walking together" and "following" end up being close to each other. If we assume basic object detection, we can resolve these ambiguities based on the class of objects performing the activities. In the next set of our experiments we separately show the classification results with and without the object label information.

Querying: To assess the efficacy of our method, we also tested it in a query-based retrieval framework. For this purpose, we used activity queries from the parking lot dataset and searched in the combined database (YouTube & VIVID). There was no overlap between the query video and the search database. The results of the experiment in the form of precision/recall table are shown in Fig. 6 and Fig. 7 where Fig. 6 demonstrates results without considering object detection and Fig. 7 demonstrates results after considering basic object detection. The poor precision rate for the case where no object detection was considered are due to the fact that certain activities, e.g. A1 and A3 (refer Fig. 5) are similar regardless of the objects performing them. When basic object detection is used, the precision rate increases considerably as seen in Fig. 7.

Choice of Parameters: Our framework has two free parameters: the energy threshold E (Section 4.1) and the decision threshold τ_D (Section 5.1).

The effectiveness of the algorithm relies on the optimum partitioning of the TAGs and the RDs into SOMs, which in turn relies on the optimal selection of Energy Threshold E. Intuitively, the threshold E signifies the strictness of incoherence between two partitions and can be formally described as:

$$E \propto G(\phi_{p1}, \phi_{p2}),$$

where ϕ_{p1}, ϕ_{p2} are the means of orientations of two partitions p1, p2 and G is the angular distance defined in Eq. (1). A higher value of E would imply merging of smaller partitions until the links of the chain have a sharp angular bend leading to high inter-incoherence. On the other hand, a lower value of E would mean that links having a small angular bend would also be divided into separate partitions. A similar logic follows for the RD vs. time plots. Currently, the thresholds are chosen by empirical experiments. Future work will consider learning these threshold values automatically from the dataset.

7. Conclusion

In this research, we presented track-based novel feature descriptors for low-level atomic and complex activity analysis in low resolution videos or far fields. We showed that temporal angular gradients of the directions in conjunction with the relative distance plots can uniquely model the complete activity spectrum. Features were divided into chains or strings of motion-words where each link was structurally coherent. These SOMs were successfully tested across various low-resolution video datasets and impressive results were obtained.

References

[1] S. Ali, A. Basharat and M. Shah, Chaotic Invariants for Human Action Recognition. IEEE ICCV, 2007.

[2] K.E Astrom and L. Morin. Random cross ratios. SCIA 1995.

[3] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. IEEE Trans. Patt. Anal. Mach. Int., 2000.

[4] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie. Behavior recognition via sparse spatio-temporal features. VS-PETS, 2005.

[5] L. Gorelick , M. Blank, E. Shechtman, M. Irani and R. Basri. Actions as Space-Time Shapes. IEEE Trans. Patt. Anal. Mach. Int., 2007.

[6] P. Turaga, R. Chellappa, V. S. Subrahmanian and O. Udrea, "Machine Recognition of Human Activities: A Survey", in IEEE Trans. Circ. Sys. Vid. Tech., 2008.

[7] J. Liu, M. Shah. Learning Human Actions via Information Maximization. IEEE CVPR, 2008

[8] J. Liu, S. Ali and M. Shah, Recognizing Human Actions Using Multiple Features, IEEE CVPR, 2008.

[9] M. Rodriguez, J. Ahmed, and M. Shah, Action MACH: A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition. IEEE CVPR, 2008

[10] F. Lv, R. Nevatia Single View Human Action Recognition using Key Pose Matching and Viterbi Path Searching. IEEE CVPR, 2007.

[11] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. ICCV, 2003.

[12] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. ICPR, 2004.

[13] E. Shechtman and M. Irani, Space-time behavior based correlation–OR–How to tell if two underlying motion fields are similar without computing them? IEEE Trans. Patt. Anal. Mach. Int., 2007.

[14] V. Parameswaran and R. Chellappa, View Invariants for Human Action Recognition. IEEE CVPR, 2003.

[15] C. Rao and M. Shah. View-invariant representation and learning of human action. IEEE Workshop on Detection and Recognition of Events in Video, 2001.

[16] N. Vaswani, A. K. Roy-Chowdhury, R. Chellappa. Activity Recognition Using the Dynamics of the Configuration of Interacting Objects, IEEE CVPR, 2003.

[17] A. Veeraraghavan, A. Roy-Chowdhury, R. Chellappa, Matching Shape Sequences in Video with Applications in Human Motion Analysis. IEEE Trans. Patt. Anal. Mach. Int., 2005.

[18] Video Verification of Identity (VIVID), www.darpa.mil/ipto/programs/vivid/vivid_approach.asp

[19] Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Computer Vision Image Understanding, 2006.

[20] Q. Fan, R. Bobbitt, Y. Zhai, A. Yanagawa, S. Pankanti, A. Hampapur. Recognition of Repetitive Sequential Human Activity, IEEE CVPR, 2009.

[21] J. Liu, J. Luo and M. Shah, Recognizing Realistic Actions from Videos "in the Wild", IEEE CVPR, 2009.

	A1: Person Entering Building	A2: Person Exiting Building	A3: Person Entering Vehicle	A4: Person Exiting Vehicle	A5: People Walking Together	A6: Vehicles Following	A7: Vehicle Starting	A8: Vehicle Stopping
Person Entering Building		1826.33	6.04	1881.67	326.37	251.41	37.32	981.23
Person Exiting Building	1826.33		1862.5	2.83	295.4	343.38	761.61	28.7
Person Entering Vehicle	6.04	1862.5		1927.13	338.04	267.47	13.65	745.71
Person Exiting Vehicle	1881.67	2.83	1927.13		314.6	356.94	659.43	112.42
People Walking Together	326.37	295.4	338.04	314.6		3.0752	412.92	317.3
Vehicles Following	251.41	343.38	267.47	356.94	3.0752		307.28	387.86
Vehicle Starting	37.32	761.61	13.65	659.43	412.92	307.28		1231.9
Vehicle Stopping	981.23	28.7	745.71	112.42	317.3	387.86	1231.9	

Figure 5: Average distance of multiple instances of interactive activities, a smaller number signifies more similarity.

	Precison	Recall	Total Fetched	True Positive	Ground Truth
U-Turn	0.71	0.75	17	12	16
Turn	0.75	0.82	12	9	11
Person Entering Building	0.8	1	5	4	4
Person Exiting Building	1	1	2	2	2
Person Entering Vehicle	0.75	1	4	3	3
Person Exiting Vechicle	1	1	3	3	3
People Walking Together	0.8	0.8	5	4	5
Vehicles Maintaing Distance	0.71	0.71	7	5	7
Vehicle Start	1	1	2	2	2
Vehicle Stop	0.67	1	3	2	2

Figure 6: Precision/Recall values for various pre-defined activities as queried in the combined dataset (with object detection).

	Precison	Recall	Total Fetched	True Positive	Ground Truth
U-Turn	0.67	0.75	18	12	16
Turn	0.47	0.73	17	8	11
Person Entering Building	0.67	1	6	4	4
Person Exiting Building	0.4	1	5	2	2
Person Entering Vehicle	0.6	1	5	3	3
Person Exiting Vechicle	0.5	0.67	4	2	3
People Walking Together	0.57	0.8	7	4	5
Vehicles Maintaing Distance	0.67	0.57	6	4	7
Vehicle Start	0.67	1	3	2	2
Vehicle Stop	0.67	1	3	2	2

Figure 7: Precision/Recall values for various pre-defined activities as queried in the combined dataset (without object detection).