Proceedings of the IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan, Oct 2009

Spatio-Temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities

M. S. Ryoo[1,2] and J. K. Aggarwal[2]
[1]Robot Research Department, Electronics and Telecommunications Research Institute, Korea
[2]Computer & Vision Research Center, The University of Texas at Austin, U.S.A.
mryoo@etri.re.kr, aggarwaljk@mail.utexas.edu

Abstract

Human activity recognition is a challenging task, especially when its background is unknown or changing, and when scale or illumination differs in each video. Approaches utilizing spatio-temporal local features have proved that they are able to cope with such difficulties, but they mainly focused on classifying short videos of simple periodic actions. In this paper, we present a new activity recognition methodology that overcomes the limitations of the previous approaches using local features.

We introduce a novel matching, spatio-temporal relationship match*, which is designed to measure structural similarity between sets of features extracted from two videos. Our match hierarchically considers spatio-temporal relationships among feature points, thereby enabling detection and localization of complex non-periodic activities. In contrast to previous approaches to 'classify' videos, our approach is designed to 'detect and localize' all occurring activities from continuous videos where multiple actors and pedestrians are present. We implement and test our methodology on a newly-introduced dataset containing videos of multiple interacting persons and individual pedestrians. The results confirm that our system is able to recognize complex non-periodic activities (e.g. 'push' and 'hug') from sets of spatio-temporal features even when multiple activities are present in the scene.*

1. Introduction

Human activity recognition, an automated detection of ongoing activities from video data, is an important problem. Semantic analysis of activity videos enables construction of various vision-based intelligent systems, including smart surveillance systems, intelligent robots, action-based human-computer interfaces, and monitoring systems for children and elderly persons. For instance, a methodology to automatically detect and distinguish suspicious and violent activities such as punching and pushing from normal activities makes smart surveillance possible. Multiple activities must be recognized and located, even when there are pedestrians and/or other interacting persons.

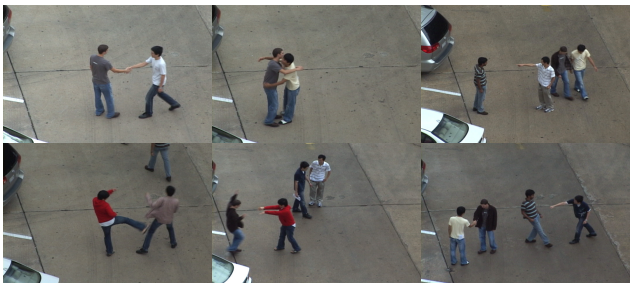


Figure 1. Snapshots of our activity videos. In contrast to the previous datasets, the videos contain several actors performing multiple interactions in the same scene. Pedestrians are also presented.

Recently, spatio-temporal feature-based approaches have been considered by several researchers [13, 3, 16, 9]. Motivated by the recent success of object recognition using local features such as SIFT descriptors [7], these approaches model a human activity as a set of sparse local descriptors directly extracted from a 3-D XYT volume (i.e. a concatenation of image frames along time axis). They do not rely on background subtraction or human body-part segmentation, and thus have been proved to be relatively immune to noise, camera jitter, changing background, and variations in size and illumination. For instance, feature-based approaches have been applied to classify videos with dynamic camera movements, such as movie scenes [6].

However, even though the previous spatio-temporal feature-based approaches have been successful on *classification* of a short video containing a single periodic action such as 'walking' and 'waving', they are limited on *recognition* (i.e. detection with localization) of complex non-periodic activities from a video containing multiple persons. In real-world applications, actions and activities are seldom periodic (e.g. 'pushing' and 'hand shaking'), and we do not

know the entire class of possible actions. In addition, multiple persons perform various interactions in the same scene, which may be organized temporally and spatially to form complex high-level activities with a hierarchical nature.

In this paper, we propose a novel video matching approach, *spatio-temporal relationship match*, which enables the recognition of complex human activities from realistic videos. The spatio-temporal relationship match is a new matching function to measure similarity between two sets of features extracted from different videos. The match explicitly compares temporal relationships (e.g. $before$ and $during$) as well as spatial relationships (e.g. $near$ and far) among extracted feature points in a 3-D XYT space, making it suitable for detecting complex non-periodic activities. Approaches utilizing the Allen's temporal predicates [1] have shown successful results on modeling human activities with complex structures [4, 11], but there has been little attempt to utilize the predicates for a space-time feature based analysis as we do in this paper. Further, our match is designed to support a scale invariant localization of the detected activities, by estimating spatial and temporal scale difference between two videos from the relations.

The hierarchical recognition of complex activities based on simpler activities is also possible with our matching. The previous space-time approaches were unsuitable for this purpose, mainly because of their inability to handle multiple actions in the same scene. Our recognition methodology has been designed and implemented to support the hierarchical process. In order to justify our approach, we test our system with the new complex dataset composed of realistic multi-person interaction videos (e.g. two persons 'hugging') that has not been tested on other space-time approaches.

In Section 2, we discuss previous works related to our paper. Section 3 provides the detailed description of the spatio-temporal relationship matching of videos. The human activity detection algorithm and the localization algorithm utilizing our relationship match is presented in the Section 4, together with a hierarchical algorithm. In Section 5, our system is tested on a complex dataset containing multiple activities. Section 6 concludes the paper.

2. Related works

Several researchers have focused on tracking persons and their body parts to recognize human actions. In general, these approaches rely on a background subtraction to segment the foregrounds (i.e. humans) and/or a methodology to estimate human body parts. They recover 2-D (or 3-D XYZ) locations and status of human body parts per frame, and model actions as a sequence of features. Dynamic time warping (DTW) algorithms and hidden Markov models (HMMs) have been popularly used to recognize actions by analyzing sequential changes in features extracted per frame (e.g. joint angles) [15]. Hierarchical approaches built upon the tracking of persons and/or their body parts have also been developed [4, 11]. They represented a complex activity as a concatenation of sub-events, and have obtained successful results.

Table 1. A table comparing properties of previous systems and our new approach. 'Structure' indicates whether the system considers organization of feature points or not; 'Localization' specifies the ability to locate where the activity is occurring; and 'Multiple activities' indicates the ability of the system to analyze videos containing multiple activities occurring simultaneously and/or sequentially. k is the number of feature vocabularies, n is the number of features, x is the number of location candidates, and l is the average number of common features vocabularies in two inputs. In general, $x > n > k > l$.

Approaches	Structure	Localization	Multiple activities	Hierarchical	Complexity
Dollar *et al.* [3]					$O(kn)$
Niebles *et al.* [9]		v	limited		$O(kn)$
Savarese *et al.* [12]	proximity only	v	limited		$O(kn + k^2)$
Scovanner *et al.* [14]	co-occur only				$O(kn + k^2)$
Wong *et al.* [16]	v	v			$O(knx)$
Our approach	v	v	v	v	$O(kn + l^2)$

Activity recognition approaches that analyze a 3-D XYT volume itself [2] or features extracted from it have gained particular interest in the past few years [13, 3, 16, 5, 9, 6]. Researchers have shown that the spatio-temporal feature based approaches are robust to noise, small camera movements, and changes in lighting conditions, overcoming the limitations of the tracking-based approaches. Dollar *et al.* [3] proposed a new local feature extractor in a spatio-temporal dimension, and applied support vector machines (SVMs) to classify videos containing each activity. They introduced a feature descriptor called 'cuboid', which captures information inside a small 3-D XYT volume patch, and represented an activity as a bag-of-features. Scovanner *et al.* [14] developed 3-D SIFT descriptors, and used SVM classifiers on co-occurrence matrices of the 3-D SIFT features to classify videos. Niebles and Fei-Fei [8] adopted static Canny edge features and the spatio-temporal features from [3], and applied part-based SVMs to classify actions per frame.

Several researchers have worked on approaches using probabilistic models. Niebles *et al.* [9] took advantage of the Dollar's feature detector, and constructed an action learning system using probabilistic latent semantic analysis (pLSA). They assumed that each feature is discriminative enough to identify likely action per feature, and applied a spatial clustering to localize actions. Wong *et al.* [16] adopted an extended version of pLSA, pLSA-ISM (implicit shape model), which considers spatial location information

for classifying videos. Savarese *et al.* [12] proposed a new matching kernel using 'correlograms', and have suggested that the consideration on the spatio-temporal proximity among features benefits the system.

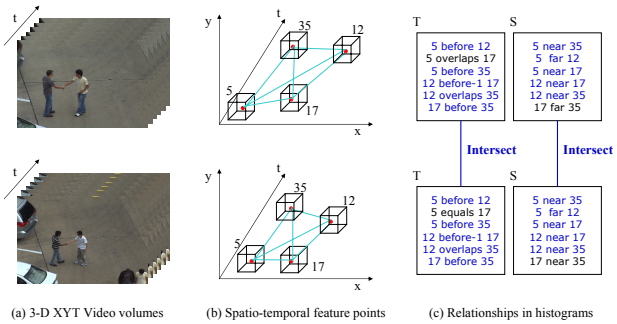
However, as pointed out by [9], since the previous probabilistic model-based approaches rely on a feature histogram to represent an activity, they inherently require a reasonably large portion of the entire video features to be those from the activity being recognized. This limits them from detecting activities when a video contains other moving objects (e.g. pedestrians), and from detecting multiple activities occurring in a same scene. One possible way for the detection is to use the sliding windows method of both space and time dimensions, but this method consumes a large amount of (if not intractable) computations.

In this paper, we propose a new spatio-temporal feature-based activity recognition methodology. Table 1 compares the abilities of the previous systems and our proposed system. The localization ability of [9, 12] is indicated as 'limited', because they are able to distinguish only a limited number of actions (2~3) in the same scene by using a spatial clustering. The main contribution of our paper is the introduction of the spatio-temporal relationship match. Up to our knowledge, there has not been any previous system which is able to address all of the above-mentioned issues.

3. Spatio-temporal relationship match

In this section, we present a novel kernel function to measure similarity between two videos, the *spatio-temporal relationship match*. A kernel function maps pairs of input data into real numbers, $K : V \times V- > R$ where V is the input space. Our kernel function serves as a likelihood measurement between two sets of feature vectors extracted from two videos containing human activities. An appropriate kernel function capturing characteristics of the activities is essential for classifying and detecting activity videos, which enables the correct recognition of the activities.

The basic idea of our spatio-temporal relationship match is to evaluate the similarity between the structures of two sets of feature points. Given a set of spatio-temporal features extracted from a video, i.e. local video patches in a 3-D XYT space, our method calculates the spatial and temporal relationships satisfied by the feature points (e.g. point f_1 is $before$ f_2, and f_1 is $near$ f_2). By comparing such relationships, the spatio-temporal relationship match measures "how many features two videos contain in common, and how many among them exhibit an identical relation" (Figure 2). The motivation behind our match is that feature points extracted from an activity video generate a particular spatio-temporal pattern in its 3-D XYT space. The major advantage of our match kernel is its efficient consideration of spatio-temporal structures among feature points. We represent the structure of the 3-D feature points as a set of pairwise relationships, and match them efficiently.



(a) 3-D XYT Video volumes (b) Spatio-temporal feature points (c) Relationships in histograms

Figure 2. An example of our spatio-temporal relationship matching process. Given two videos (i.e. (a)), the system extracts feature points and analyze their pair-wise relations as presented in (b). The relationship histogram is computed per input video as described in (c), and they are intersected to measure their similarity.

3.1. Features and their relations

A spatio-temporal local feature in a 3-D XYT dimension (i.e. a feature point) contains two types of information: appearance information of the video patch, and its location information in the 3-D space. A spatio-temporal feature extractor (e.g. [3, 14]) detects each interest point locating a salient change in a video volume, and constructs a descriptor summarizing a small video patch around it. That is, a feature f is composed of two parts, described as $f = (f^{des}, f^{loc})$ where f^{des} is an appearance descriptor and f^{loc} is a 3-D coordinate of the feature.

We desribe the appearance part of features with k possible vocabularies. We group spatio-temporal features into k clusters while ignoring the location part, so that each feature corresponds to one of k possible types. More specifically, we use k-means algorithm to cluster all f based on their f^{des} value. Let an input v be a set of features, f_1, ..., f_n. If we denote the center of the ith cluster as $mean_i$, then each f in v goes into one of the k histogram bins as follows:

$$H_i(v) = \{f \mid f \in v \wedge i = argmin_j \, ||f - mean_j||^2\} \quad (1)$$

where $||f - mean_j||$ is a distance between f and $mean_j$, the Euclidean distance in our case. As a result, we have decomposed the set v into k subsets (i.e. types), $H_1(v)$, ..., $H_k(v)$, based on their appearance.

Next, we measure the spatio-temporal relationships among feature points, f_1^{loc}, ..., f_n^{loc}. We consider both temporal relationships and spatial relationships, and use a set of pair-wise predicates to describe the relations. That is, for all possible pairing of f_a^{loc} and f_b^{loc} where $1 \leq a, b \leq n$, we analyze how two points are located in the XYT space.

Temporal ordering of feature points is particularly important, and we adopt the Allen's temporal predicates to describe temporal relations: $equals$, $before$, $meets$,

$overlaps$, $during$, $starts$, and $finishes$ [1]. These predicates describe relationships between two time intervals. A spatio-temporal feature descriptor tends to describe a 3-D volume patch instead of a single point (e.g. Figure 2 (b)), meaning that it forms an interval when projected to the time axis. Further, since a human activity is seldom instantaneous and takes certain duration, the interval representation of features is necessary for our matching to hierarchically recognize activities by treating simpler activities as features.

For each feature location f^{loc}, we compute the starting time and the ending time (f^{start}, f^{end}) of the volume patch associated with it, and describe their relations using the temporal predicates:

$$\begin{aligned}
equals(f_a, f_b) &\iff f_a^{start} = f_b^{start} \; and \; f_a^{end} = f_b^{end} \\
before(f_a, f_b) &\iff f_a^{end} < f_b^{start} \\
meets(f_a, f_b) &\iff f_a^{end} = f_b^{start} \\
overlaps(f_a, f_b) &\iff f_a^{start} < f_b^{start} < f_a^{end} \\
during(f_a, f_b) &\iff f_a^{start} > f_b^{start} \; and \; f_a^{end} < f_b^{end} \\
starts(f_a, f_b) &\iff f_a^{start} = f_b^{start} \; and \; f_a^{end} < f_b^{end} \\
finishes(f_a, f_b) &\iff f_a^{end} = f_b^{end} \; and \; f_a^{start} > f_b^{start}
\end{aligned}$$

We also consider the reverse predicates of the above predicates except $equals$. We say that a predicate for two feature points $pred(f_a, f_b)$ is true if and only if the condition corresponding to the $pred$ is satisfied by f_a^{loc} and f_b^{loc}. In case of point-wise features, only three temporal predicates, $equals$, $before$, and $before^{-1}$, are considered.

Similarly, spatial predicates describing spatial distance between two feature points are designed. The predicate $near$ indicates that two feature points are spatially closer than a certain threshold, and the predicate far implies that the two points are not close. $xnear$ (or $ynear$) measures whether x (or y) coordinates of two features are near.

$$\begin{aligned}
near(f_a, f_b, threshold) &\iff dist(f_a, f_b) \leq threshold \\
xnear(f_a, f_b, threshold) &\iff dist(f_a^x, f_b^x) \leq threshold \\
&\quad \wedge \neg \, near \\
ynear(f_a, f_b, threshold) &\iff dist(f_a^y, f_b^y) \leq threshold \\
&\quad \wedge \neg \, near \wedge \neg \, xnear \\
far(f_a, f_b, threshold) &\iff \neg \, near \wedge \neg \, xnear \wedge \neg \, ynear
\end{aligned}$$

The set of satisfied pair-wise relationships provides us a compact and efficient representation of the spatio-temporal structure of the 3-D feature points. The intention is that two videos containing an identical activity will contain similar features having similar pair-wise relations.

3.2. Spatio-temporal relationship match kernel

Our spatio-temporal relationship match kernel is a histogram-based match kernel, which measures the similarity by constructing histograms and intersecting them. We introduce a new histogram capturing both appearance and relationship information of a video: the *relationship histogram*. A relationship histogram is a 3-dimensional histogram where the dimensions correspond to $featuretype \times featuretype \times relationship$. Each bin of a relationship histogram is designed to contain designated pairs of two feature points from a video: Let $R(v)$ be a relationship histogram of an input v. Then, a feature pair (f_a, f_b) is in a relationship histogram bin $R^{rel}_{(i,j)}$, if and only if f_a is of the appearance type i, f_b is of j, and they satisfy the relationship rel (i.e. $rel(f_a, f_b)$ is $true$).

$$\begin{aligned}
R^{rel}_{(i,j)}(v) = \{(f_a, f_b) \mid f_a \in H_i(v) \\
\wedge f_b \in H_j(v) \wedge rel(f_a, f_b) \wedge i < j\}
\end{aligned} \tag{2}$$

That is, each bin collects pairs with two particular types which satisfy a specified relationship. We assign a pair to a bin only when $i < j$, in order to avoid an identical pair to appear again in its reverse bin with their orders reversed.

There are two types of relationship histograms: a temporal relationship histogram, and a spatial relationship histogram. For each video v, we construct one temporal relationship histogram $T(v)$ and one spatial relationship histogram $S(v)$. A bin of a temporal relationship histogram collects pairs with a specific temporal relationship, while that of a spatial relationship histogram collects pairs of a spatial relation. Since only one temporal relationship and one spatial relationship are satisfied per a feature pair, each histogram divides entire pairs of feature points in a video into $k^2 \cdot r$ subsets where r is the number of relationships.

$$\begin{aligned}
T^{trel}_{(i,j)}(v) = \{(f_a, f_b) \mid f_a \in H_i(v) \\
\wedge f_b \in H_j(v) \wedge trel(f_a, f_b) \wedge i < j\} \\
S^{srel}_{(i,j)}(v) = \{(f_a, f_b) \mid f_a \in H_i(v) \\
\wedge f_b \in H_j(v) \wedge srel(f_a, f_b) \wedge i < j\}
\end{aligned} \tag{3}$$

where $trel$ is one of 13 temporal relationships and $srel$ is one of 4 spatial relationships.

Our match kernel, K_R, measures the similarity between two inputs $v1$ and $v2$, by intersecting temporal and spatial histograms (i.e. $T(v1) \cap T(v2)$ and $S(v1) \cap S(v2)$). An intersection of two histograms contains a pair of feature points that is presented in both inputs with a similar appearance and relationship. Thus, counting the number of pairs in the intersections of histograms from two inputs provides us how many pair-wise relations two videos share. Figure 2 shows an example process of our matching.

The function to count the number of pairs in an intersection between two histogram bins $B1$ and $B2$ is as follows:

$$I(B1, B2) = min(|B1|, |B2|) \tag{4}$$

Using the histogram intersection function I, we define the match kernel K_R which counts the number of pairs in the entire bins of the intersection.

$$\begin{aligned}
K_R(v1, v2) = \sum_{i=1}^{k} \sum_{j=1}^{k} [\sum_{trel} I\left(T^{trel}_{(i,j)}(v1), T^{trel}_{(i,j)}(v2)\right) \\
+ \sum_{srel} I\left(S^{srel}_{(i,j)}(v1), S^{srel}_{(i,j)}(v2)\right)]
\end{aligned} \tag{5}$$

Since we only need to consider histogram bins with feature types common in both inputs, our relationship intersection can be computed in $O(l^2)$ where l is the number of common feature types in $v1$ and $v2$ ($k > l$).

What we must note is that our kernel is expressed only in terms of the histogram intersection functions. Based on the fact that the histogram intersection is a Mercer's kernel [10], and the Mercer's condition is closed under addition, we know that our match kernel is a Mercer's kernel. This guarantees the optimal solution for kernel-based algorithms using convex optimization, including SVMs.

4. Human activity recognition

In this section, we present methodologies for detection and localization of human activities. Our system takes advantage of the relationship match kernel presented in the previous section, detecting activities by comparing videos (Subsection 4.1) and localizing them by estimating the starting and ending locations of the activities (Subsection 4.2). We also present how our algorithm is applied hierarchically for the recognition of high-level activities in Subsection 4.3.

4.1. Activity detection and partial matching

We detect/match human activities by comparing videos. Our system decides whether the testing video contains an activity or not, by measuring the similarities between the video and other training videos containing the activities.

Our system maintains one training dataset D_α per activity α. Each training set is composed of several videos of different persons performing one activity with various scales and backgrounds. For each training video, we extract a set of features only from a region around the person performing the action, since it may contain many actors. Sets of features extracted from training videos D_α are maintained, so that they can be compared with a testing video.

Given a testing video v_{test}, our system calculates the similarities between v_{test} and all elements of D_α using our spatio-temporal relationship match kernel. In general, the number of features in a training video is significantly smaller than that of features in a testing video. A training video contains only one activity while a testing video may contain many, suggesting that the ability to perform partial matching is essential. Since our spatio-temporal relationship match kernel is a histogram intersection kernel, our match supports partial matching. With the assumption that the number of feature points are similar in all training videos, our spatio-temporal relationship match is able to provide an optimal decision boundary for the detection, even when a testing video contains more than one activity.

In order to compensate for the difference in the number of feature points among training videos and to make the system learn correct decision boundaries, we normalize each of the similarities computed by our matching. Let D_α^m denote features extracted from mth training video in the set D_α. The match between D_α^m and the testing video v_{test}, $K_R(D_\alpha^m, v_{test})$, is divided by the number of relationships in the training data D_α^m. That is, we are measuring the ratio of relationships that are common in both inputs (i.e. an interaction) and those in the training input:

$$match\_lk(D_\alpha^m, v_{test}) = K_R(D_\alpha^m, v_{test}) / K_R(D_\alpha^m, D_\alpha^m) \quad (6)$$

For each training dataset D_α, the system computes a set of training videos *close* to the testing video, C_α:

$$C_\alpha(v_{test}) = \{D_\alpha^m \mid match\_lk(D_\alpha^m, v_{test}) > th\} \quad (7)$$

where th is a threshold value between 0 and 1 that has to be learned. Based on the number of elements in $C_\alpha(v_{test})$ (i.e. number of similar videos in the training set), the system decides whether the video v_{test} contains the activity α.

4.2. Localization

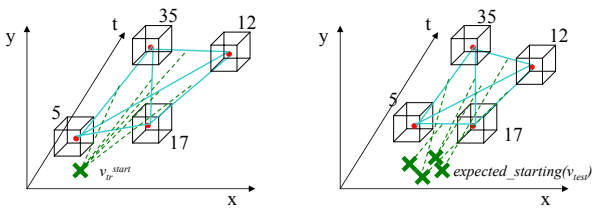
Once our system decides that a video contains an activity, the system localizes it. The system detects occurring locations of the activity (note that a video may contain multiple executions of the same activity), by searching for the activity's spatial coordinates, its starting time, and its ending time. We develop a localization algorithm based on voting: For each training video v_{tr} in $C_\alpha(v_{test})$, our system computes the intersection of relationship histograms of v_{tr} and v_{test} by performing the spatio-temporal relationship match. Each pair of features in the intersection votes for the expected starting and ending locations of the activity in v_{test}, estimating them based on the training data.

We assume that the location of the activity in each training video is provided along with the feature points extracted (i.e. labeled training data). The starting location $v_{tr}^{start} = (x, y, t^{start})$ and the ending location $v_{tr}^{end} = (x, y, t^{end})$ is specified for each training video, and the goal is to find those of the activity in the testing video.

Our system calculates the relative position of the activity's starting location (or the ending location) from the center of each pair in a training video. This tells us where the pair thinks the starting location is. We normalize the relative position by the scale of the pair (i.e. distance between two elements of the pair), in order to make our algorithm scale invariant. Since there are multiple pairs that have identical feature types, we compute and maintain the set of normalized relative positions per a pair of feature types as follows:

$$\begin{aligned} &start\_relative_\alpha(i,j) = \\ &\{d \mid \exists v_{tr}\ \exists f_a\ \exists f_b : d = \frac{v_{tr}^{start} - (f_a^{loc} + f_b^{loc})/2}{\sqrt{(f_a^{loc} - f_b^{loc})^T (f_a^{loc} - f_b^{loc})}} \\ &\wedge v_{tr} \in C_a(v_{test}) \wedge f_a \in H_i(v_{tr}) \wedge f_b \in H_j(v_{tr})\} \end{aligned} \quad (8)$$

When computing the $f_a^{loc} + f_b^{loc}$ and $f_a^{loc} - f_b^{loc}$, we only consider center locations of the feature volumes f_a and f_b. Each of v_{tr}^{start}, f_a^{loc}, and f_b^{loc} has three dimensions (x, y, t),



(a) A starting location of a training video (b) The location voting in a testing video

Figure 3. An example of our localization process. (a) shows a specified starting location of a training video, and (b) shows an example voting for a testing video.

and thus each d is a 3-dimensional vector. The normalized relative position of the ending location $end\_relative_\alpha(i,j)$ is computed similarly.

Next, our system computes the expected position of the activity's starting (or ending) location in the testing video based on the $start\_relative$ (or $end\_relative$) computed with each training video. The idea is that the normalized distance between the center of a pair and the starting (or ending) location of an activity will be similar for all videos containing the activity. By scaling the normalized relative positions specified in $start\_relative$ for each feature pair in the testing video, and by adding the relative position to the center of the pair, the system is able to calculate the expected starting location of the activity in the testing video. Each pair generates an expected location, constructing a set of expected locations as follows:

$$expected\_starting_\alpha(v_{test}) = \{s \mid \exists f_a\ \exists f_b\ \exists i\ \exists j : f_a \in H_i(v_{test}) \wedge f_b \in H_j(v_{test}) \wedge s = (f_a^{loc} + f_b^{loc})/2 + \sqrt{(f_a^{loc} - f_b^{loc})^T (f_a^{loc} - f_b^{loc})} \cdot start\_rel_\alpha(i,j)\} \quad (9)$$

where we compute $expected\_ending_\alpha(v_{test})$ similarly. Figure 3 shows an example localization.

The localization decision is made by combining all votes in the set $expected\_starting_\alpha(v_{test})$. We divide the entire 3-D XYT space into several bins, and count the number of votes inside each bin. Bins with local maxima are selected as starting (or ending) time candidates, only when they have sufficient votes as compared to the number of feature pairs in the corresponding training video. The locations are computed for each training video, and the system concatenates the locations obtained by all training videos. The computational complexity of our localization algorithm per video is $O(l^2 + x)$ where x is the number of 3-D bins.

4.3. Hierarchical recognition

As pointed out by several researchers [1, 4, 11], many human activities are composed of several atomic-level actions organized hierarchically. In order to make our system recognize complex high-level activities, our detection and localization algorithms have been designed so that they can be applied repeatedly, enabling the hierarchical recognition.

Our system first detects simple atomic actions (e.g. 'arm stretching' and 'arm withdrawing') from a set of input features. Once detected, our localization algorithm searches for its starting time, ending time, and spatial location, as well as the width and height of the bounding box containing the features spatially. As a result, a localized atomic action forms a 3-D XYT cuboid with a certain width, height, and depth describing where the action is occurring. Treating the localized actions as new features, our recognition algorithm is applied hierarchically.

The key is that our recognition algorithm has an ability to correctly detect and localize (instead of classifying) human activities from a video where multiple persons and actions are present. Such an ability makes the encapsulation of recognized actions into new 'features' possible, enabling hierarchical application of the algorithms. For instance, a two person interaction of a 'hand-shake' can be recognized by matching relationships formed with atomic actions 'arm stretching' and 'arm withdrawing' of two persons: if we denote stretching and withdrawing actions of persons 1 and 2 as $st1$, $st2$, $wd1$, and $wd2$, the most typical relationships among them include $st1\ before\ wd1$, $st2\ before\ wd2$, $st1\ equals\ st2$, $wd1\ equals\ wd2$, and so on.

5. Experiments

5.1. Dataset

In order to test our system to recognize multiple high-level activities from a video, we have constructed a new dataset. Our dataset contains six types of two-person interactions, which we define to be composed of 10 types of non-periodic atomic-level actions. Shake-hands, point, hug, push, kick, and punch are the six classes of interactions. Stretch arm, withdraw arm, stretch leg, lower leg, and shift forward, of left and right directions are the 10 types of atomic actions composing the interactions. The dataset is composed of 10 sets, where each set contains videos of a pair of different persons performing all six interactions. In sets 1 to 4, only two interacting persons appear in the scene. In sets 5 to 8, both interacting persons and pedestrians are present in the scene. In sets 9 and 10, several pairs of interacting persons execute the activities simultaneously. Each set has a different background, scale, and illumination. 6 participants performed activities with 10 different clothing conditions. Total of 60 interactions and more than 180 atomic actions are in the entire dataset.

The dataset is composed of several 'unsegmented' videos; a video in the dataset contains multiple executions of interactions and atomic actions, occurring sequentially and concurrently. On average, each video contains about two interactions composed of several atomic actions. Figure 1 shows example snapshot images of our dataset. As shown in the snapshots, not only the interactions of target

persons but also irrelevant pedestrians are present in the videos. Each interaction or atomic action is labeled with its type, starting location, and ending location, so that they can be used either for training or for testing.

Even though the Weizmann and KTH datasets [2, 13] have been popularly used for measuring classification accuracy of action recognition systems, the datasets were limited in the sense that each of their videos contain a single periodic action. All actions provided are periodic, except the 'bend' action in the Weizmann dataset. The dataset used in [11] contains similar activities to our dataset, but their videos were taken in an indoor environment with a fixed view point, and with only two persons in a scene. Our videos contain multiple persons and activities with various conditions from an aerial image-like view point, which prevents the previous systems from directly being applied.

5.2. Results

We have conducted three types of experiments to verify the advantages of our system. In the first experiment, we test our system on the action classification task using a public dataset, the KTH dataset [13]. Next, we compare our system with the previous systems on an atomic action detection and localization task, using our new dataset. In the third experiment, we use our system to recognize six types of complex human-human interactions. The experiments confirm that our system is able to analyze realistic videos.

We have implemented the matching, detection, and localization algorithms presented in the previous sections. We took advantage of the 'cuboid' spatio-temporal feature extractor developed by [3]. The samples from the extracted features are clustered into 500 types based on their appearance for the matching (i.e. $k = 500$), similar to [9]. For the localization voting, the entire 3-D XYT space is divided into volume patches with the size of $10 * 10 * 5$.

In the first experiment, we have applied our spatio-temporal relationship match for the classification of simple actions in the KTH dataset. Even though the focus of our methodology is less on the classification of simple actions (i.e. our contribution is on the detection and localization of complex human activities that previous systems did not attempt), we show that the classification performance of our system is comparable to the other state-of-the-arts systems designed for the action classification problem. Table 2 compares the accuracies of our system using N-nearest neighbors ($N = 9$) and other previous systems under two different experimental settings. [16] and [5] are not directly comparable to our system, because they are using non-trivial settings as [6] have mentioned. Among the systems using the same features (i.e. cuboids) [3, 9, 12], our system obtained the best accuracy.

For the second and the third experiments, we have used our new dataset. We have randomly chosen two among 10 sets to form a training set, and used the other sets for the testing. We intentionally made the system to use a small number of training videos, making the problem difficult, in order to verify our system's ability to learn the activities on a realistic environment where only a limited amount of training examples are available.

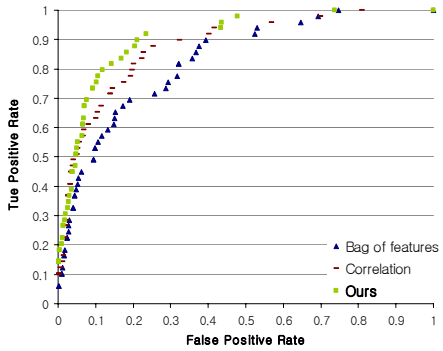


Figure 4. ROC graphs of the atomic action detections. Blue rectangles are accuracies of the system using bag-of-features [3], red '-'s are that of the system utilizing the feature correlation information similar to [14], and green triangles are that of our system. All systems use 'brightness' cuboid features. We are able to observe that our system constantly performs about 10˜20% superior to the previous systems for false positive rates 0.1˜0.3. The area under ROC are 0.83, 0.87, and 0.91 respectively.

In the second experiment, we compared our system with the previous systems on detecting 10 atomic actions from segmented videos. Among the entire testing set, we have randomly chosen 50 video segments where each of them contains only one atomic action. In addition, 20 video segments which do not contain any actions have been tested. The task is to decide whether each video contains an action or not (i.e. the detection task described in 4.1). The binary detection has been done for each type of atomic actions, and the performance has been averaged. Figure 4 compares the accuracies of our system and other systems using the same spatio-temporal features. The accuracies with respect to the false positive rates are specified. The graph suggests that our system detects simple non-periodic actions more accurately than those using the same features. The localization accuracy of our system was 0.9 on average.

Finally, we verify that our system is able to recognize multiple high-level activities from continuous videos, which has not been attempted by previous systems. The six types of human-human interactions have been recognized hierarchically using the atomic action detection results, correctly matching, detecting, and localizing multiple occurrences. Figure 5 shows an example of the recognition results of our system, and Table 3 illustrates the performance of our system. We are able to observe that our system recognizes interactions with a good accuracy, even though only few training examples are provided. The recognition

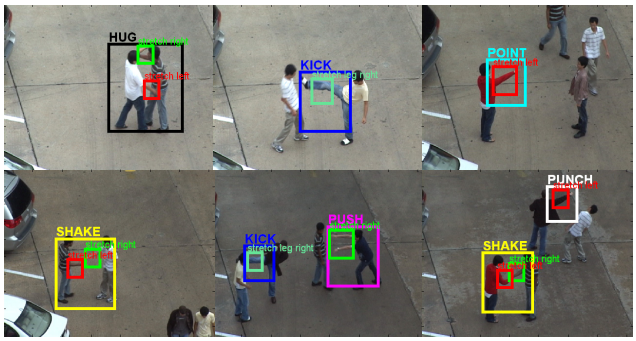


Figure 5. Example snapshots of recognition results.

accuracy of the 'punch' was relatively low, since its duration and the number of atomic actions composing it was too small to distinguish them from other activities. The results suggest that our system is able to recognize non-periodic human-human interactions even when the videos contain other interacting persons and/or pedestrians. Our system successfully recognizes activities sharing same atomic actions based on the actions' spatio-temporal relations.

6. Conclusion

We have presented a human activity recognition methodology, which is designed to detect and localize complex activities from realistic videos. Our spatio-temporal relationship match measures a similarity between two activity videos by considering structures among spatio-temporal features. We have presented a detection and a scale-invariant localization algorithm for locating multiple occurrences of activities. Through the experiments with our new dataset, we have confirmed that our proposed system is able to recognize complicated human activities hierarchically.

Ack.: This work was supported partly by the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Evaluation Institute of Industrial Technology (KEIT) [2008-S-031-01, Hybrid u-Robot Service System Technology Development for Ubiquitous City], and partly by Texas Higher Education Coordinating Board under award no. 003658-0140-2007.

Table 2. Performance comparisons of the approaches tested on the KTH dataset. The left of the 'classification accuracy' shows systems' performances with the 16 training / 9 testing setting [13, 6], while the right shows those with the 25-fold leave-one-out cross validation setting [3]. The 'performance increase' indicates the amount of the accuracy increased, compared to the baseline method (i.e. an identical approach with bag-of-words paradigm) using the same features. Our baseline is [3], using cuboid features.

System	Classification accuracy	Performance increase
Laptev *et al.* [6]	91.8 / - %	+2.1%
Ours	**91.1 / 93.8 %**	**+12.6%**
Savarese *et al.* [12]	- / 86.8 %	+5.6%
Niebles *et al.* [9]	- / 81.5 %	+0.3%
Dollar *et al.* [3]	- / 81.2 %	-
Schuldt *et al.* [13]	71.7 / - %	-

Table 3. Recognition accuracy of six complex human-human interactions from the new dataset.

Interaction	Recognition accuracy	False positive rate
shake hands	0.750	0.088
hug	0.875	0.075
point	0.625	0.025
punch	0.500	0.213
kick	0.750	0.138
push	0.750	0.125
total	0.708	0.110

References

[1] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[3] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on VS-PETS*, pages 65–72, 2005.

[4] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *CVIU*, 96(2):129–162, 2004.

[5] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[6] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[7] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[8] J. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

[9] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3), Sep 2008.

[10] F. Odone, A. Barla, and A. Verri. Building kernels from binary strings for image matching. *IEEE T Image Processing*, 14(2):169–180, Feb 2005.

[11] M. S. Ryoo and J. K. Aggarwal. Semantic representation and recognition of continued and recursive human activities. *IJCV*, 82(1):1–24, April 2009.

[12] S. Savarese, A. DelPozo, J. Niebles, and L. Fei-Fei. Spatial-temporal correlatons for unsupervised action classification. In *WMVC*, 2008.

[13] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004.

[14] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM MULTIMEDIA*, pages 357–360, 2007.

[15] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE T CSVT*, 18(11):1473–1488, Nov 2008.

[16] S.-F. Wong, T.-K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. In *CVPR*, 2007.