Distant Human Interaction Recognition with Kinect

Chengzhang Qu and Dengyi Zhang
Computer School, Wuhan University, Wuhan, P. R. China
Email: quchengzhang@whu.edu.cn; dyzhangwhu@163.com

Yuewei Lin and Song Wang
Department of Computer Science and Engineering University of South Carolina, Columbia, SC, USC
Email: lin59@email.sc.edu; songwang@cec.sc.edu

Abstract—**Detecting human interactions in a public place where no physical touch occurs has important applications in many surveillance tasks. In this paper, we explore the possibilities to automatically detect such distant human interactions without recognizing the specific human actions. Specifically, we use a highly simplified formulation of the interaction in this paper: 1) when a person does not interact with others, he always performs a non-interactive action that is largely periodical, such as walking, 2) when two persons interact with each other, they both perform a same short-duration interactive action, such as waving hand, that are different from their non-interactive actions. Based on this formulation, we develop a new approach to localize the subvideos that describes the interactive actions from Kinect videos, which provide both the RGB and depth information. We then develop a new approach to compare the pose and kinematic features in these subvideos (from different people) to see whether they describe a same interactive action. Without any supervised learning and action recognition, the proposed approaches are not limited to a specified set of interactive and non-interactive actions. In the experiments, we justify the performance of the proposed approaches on 100 Kinect videos with 10 different interactive actions.**

Index Terms—**distant human interaction, abnormal detect, dynamic time warping**

I. Introduction

When people meet in public places, they may interact with each other, either with physical touch, such as shaking hands and hugging, or without physical touch, such as waving hand and nodding. Detecting such interactions has important applications in many surveillance tasks, such as security monitoring in an airport or other public areas. Typically, the human interactions without physical touch, which we call ``distant human interactions", are much more challenging because they cannot be determined by simply examining the spatial relations between the involved persons. In this paper, we explore this problem by using the Kinect videos as input, from which human 3D poses can be more accurately detected than from the traditional RGB videos, with the additional depth information. For the remainder of this paper, we simply use “human interaction” to mean “distant human interaction” when there is no specific indications.

There are many different kinds of interactions between two persons. Other than waving hands and nodding, Japanese usually bow to each other when they meet. Monks in Asia usually clap to each other. Western gentlemen may take off the hat and then nod to each other. Considering people from different races, of different genders, with different ages, it is difficult to exhaustively collect and model all possible interactions and then use them to guide the interaction detection. In addition, whether a special human action is part of an interaction is also dependent on both the temporal and spatial context information. For example, a person standing still for a long time by himself is not an interaction. However, if two persons walk toward each other and then suddenly stop and stand still for a while simultaneously, this same action of standing still in this context may indicate an interaction. Therefore, it is also difficult to achieve the interaction detection by only recognizing special actions of individual persons. The general goal of this paper is to develop an unsupervised approach for detecting human interactions without recognizing the special actions involved in the interaction. This way, we expect our approach can be used to detect any interactions, including both seen and unseen ones.

As an exploratory study, we consider the following simplified model for human interaction in this paper.

- Each person’s action in a video is made up of a non-interactive action, when he is not interacting with others, and some interactive actions, when he is interacting with others.
- For each person, the non-interactive action is largely periodical, such as walking with a fixed speed. The interactive actions of a person are different from his non-interactive actions in terms of pose changes. In addition, we assume that in a video, the interactive actions of a person are short-duration events, compared to the non-interactive actions.
- When two persons interact with each other, they perform the same interactive actions, e.g., both of them bow to each other, or both of them wave

doi:10.4304/jcp.9.9.2091-2099

hands to each other. Note that this is a highly simplified assumption since in practice two persons may perform different actions in interacting with each other.

Based on this model, we focus on addressing the following two major problems to justify the feasibility of detecting human interactions without recognizing specific interactive actions: (P1) temporally partition a video to identify subvideos that describe the interactive actions of a person, (P2) compare the subvideos of interactive actions from different people and match the ones with the same underlying interactive action. In (P2), we match the subvideos only based on the similarity of their pose and kinematic features. In practice, we can easily include the temporal coincidence (i.e., the interactive action of two involved persons should occur at the same time) and the spatial coincidence (the persons who are interacting should face each other) to improve the interaction detection accuracy.

One issue in the experiment setting is that the current Kinect and its SDK can only track the pose of one human in a small range (in practice, a Kinect can well track the 3D human poses in the range from 1.2 meters to 3.5 meters) and with good frontal views of each person. It is very difficult to use a fixed Kinect to collect a video with two or more persons who are walking toward and interacting with each other. In practice, this can be easily solved by installing multiple Kinects that are facing different directions and covering different areas. In the ideal case, we can assume that, at any time each person can be caught by at least one Kinect with good frontal views. With these considerations, in our experiments we adopt the following simplified settings for experiments: a) we collect Kinect videos only for individual persons (with interactive and non-interactive actions), and (b) we move the Kinect (in a cart) when the person is moving to make sure the person is facing the Kinect and is located in the range of the Kinect. This simplified settings do not affect the research scope of the above two major problems: we study the temporal partitioning of each video (now each video only contains one person) for subvideos of interactive actions and then compare subvideos from different videos (i.e., different persons) to identify the matched interactive actions.

The remainder of the paper is organized as follows. Section II briefly overviews the related work. Section III describes the proposed algorithm on partitioning a video for the subvideos of interactive actions. Section IV introduces the similarity measure of subvideo matching for identifying same interactive actions from different persons. Section V describes the experiment results and Section VI concludes the paper.

II. Related Works

Many methods have been recently proposed to model the multiple-people interactive actions in video sequences, which are related to the proposed work. However, the methods developed in these previous work treat the interaction detection as a recognition problem [1]–[3] and usually require a set of training samples and a supervised learning procedure for recognizing an interactive action. Recently, Zhou et al. [4] and Ni et al. [5] use trajectory analysis to describe different group activities, which are usually based only on the relative loations and motions of the involved people without analyzing their body poses and pose changes over time. The major difference between the proposed work in this paper and these previous methods are that we consider the human poses for finding the matched interactive actions and do not use a supervised learning for recognizing the specific actions. In addition, we use Kinect videos for facilitating the pose extraction and tracking.

There are rich literatures on human activity recognition from video sequences. Recently, Aggarwal and Ryoo conducted a comprehensive review of human activity analysis [6], in which various spatial and temporal features are used. Ke et al. [7], [8] proposed models for action recognition in which the input video sequence is treated as a 3D volume and some local volumetric features are extracted. In [9], [10], local interest point descriptors are detected and used for human activity recognition. In [11], [12], the involved activity agents, such as persons, are first detected and their relations are then modeled for recognizing the underlying human activities. There are also many models have been developed to describe the identified features and agents for human activity recognition. For example, prior work has used Hidden Markov Model (HMM) to describe and distinguish the dynamics underlying different human activities [13]. In [14], Bayesian networks, together with Markov chain Monte Carlo algorithm, are used to recognize bicycle related activities. In [15], a hierarchical probabilistic latent model is developed to represent the behavior pattern. In [16], probabilistic analysis, such as stochastic-context grammars, is designed for modeling human activities in a hierarchical way. Most of these methods are focused on recognizing a small set of different human actions or activities, while in this paper, our goal is to detect distant human interaction by not limited to a few specific actions.

The proposed step of identifying subvideos of interactive actions from a video shares some similarity to the problem of anomaly detection from a video, if we treat interactive actions as abnormal actions and the non-interactive action as a normal action. In [17]–[19], supervised learning is adopted to train models for both normal and abnormal actions. In [20]–[22], unsupervised learning on the annotated training data is used for anomaly detection. Mehran et al. [23] analyze the optical flow extracted from a video to detect possible interactions in a crowd. In [24], Mahadevan et al. propose a model for anomaly detection in crowded scenes by using a dynamic saliency measurement. Cui et al. [25] propose to detect the abnormal action of a group of people by using interaction energy potential. Different from the proposed method, all these models only consider the relative location and location changes of the people without considering their detailed poses.

III. Video Partitioning for Subvideos of Interactive Actions

In this section, we develop an algorithm to temporally partition an input video to localize subvideos that describe interactive actions. As introduce above, there is only one person present in a video, which consists of a largely periodical non-interactive action and short-duration interactive actions. We focus on analyzing the human pose and kinematic features for distinguishing these two kinds of actions. Therefore, with an input Kinect video, we first apply the recently released Kinect SDK [26] to locate the person and his pose. Specifically, the Kinect SDK can track the 3D position of 20 human-body joints over time, as shown in Fig. 1. Prior research has shown that different human actions can be well described by the body-joint angles and the temporal change of these angles. In this section, we use the following 21 joint angles for video partitioning:

- 11 upper-body joint angles: one for neck, two for elbows (left and right), two for wrists (left and right), six for shoulders (left and right). For each shoulder, we compute the three angles between the shoulder-elbow line and the sagittal plane, the coronal plane and the transverse plane, respectively [27].
- 10 lower-body joint angles: two for knees (left and right), two for ankles (left and right), three for hip left, and three for hip right. For either hip left or hip right, we compute the three angles between the its connection line to the knee and the sagittal plane, the coronal plane and the transverse plane, respectively [27].

For example, by computing a shoulder angle at each frame, we construct a 1D time-varying signal for this angle as shown in Fig. 2(a). In this paper, we denote these angles as $f(t)=\{f_1(t), f_2(t),..., f_K(t)\}$ with $K=21$ being the number of considered angles and $t = 0, 1, \cdots, T-1$ being the index of the frames in the video. In the following, we use $t_1 : t_2$ to represent a video segment starting from frame t1 and ending at frame $t_2 - 1$, and use $f_i(t_1:t_2)$ to represent the subsequence of $f_i(t)$ on the video segment $t_1:t_2$, for $t_1<t_2$.

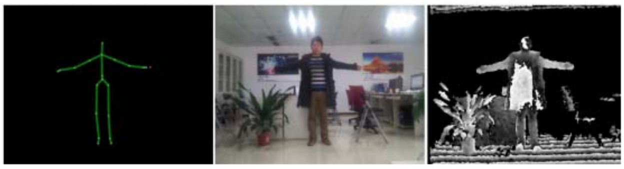

Figure 1. Example of kinect 20 joints skeleton, color, depth information

A. Period-Length Estimation

As discussed in Section I, each video is dominated by a largely periodical non-interactive action, such as walking. Therefore, the angle signals $f(t)$ is largely periodical without considering the short-duration subvideos of interactions. We first estimate the period length of the non-interactive action in a video by analyzing the dominating frequency of $f(t)$.

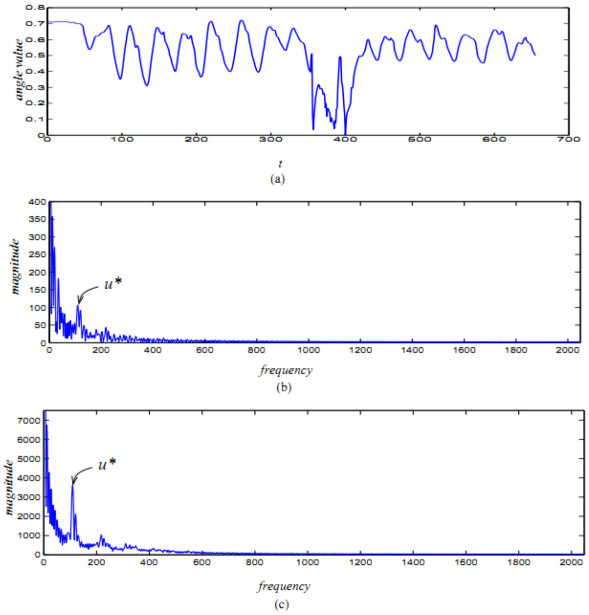

Figure 2. An illustration of period-length estimation. (a) The 1D signal of a right shoulder angle. (b) Frequency response (magnitude) of the signal shown in (a). (c) Combined frequency response (magnitude) from all K angle signals and the estimated frequency u^*.

Specifically, we first use FFT (Fast Fourier Transform) to compute the frequency response of each angle signal $f_i(t)$, i = 1, 2, · · · ,K

$$F_i(u)=\sum_{t=0}^{N-1} f_i(t)e^{-j2\pi u\frac{t}{N}}, u=0,1,...,N-1, \qquad (1)$$

where j $=\sqrt{-1}$ and $N>T$ is the length of FFT. The angle signal $f_i(t)$ is zero-padded to length N to avoid the aliasing in the frequency response. In our implementation, we pick N to be a power of 2 to make FFT more efficient.

From the frequency magnitude $|F_i(u)|$ $,u=0,1,...,N-1$, we find the local maximum at u^* # other than the zero frequency as the frequency of the non-interactive action, as shown in Fig. 2(b). This way, the period length of the non-interactive action can be estimated by $T_c=\frac{N}{u^*}$. However, this estimation of the period length is vulnerable to noise when using only one angle signal. Given that all K angle signals collectively describe a non-interactive action, they should have the same u^*. In this paper, we combine the frequency response of all K angle signals to improve the robustness of period-length estimation. Specifically, we calculate a combined frequency magnitude $\sum_{i=1}^{K}|F_i(u)|$ $,u=0,1,...,N-1$ and then find the local maximum at

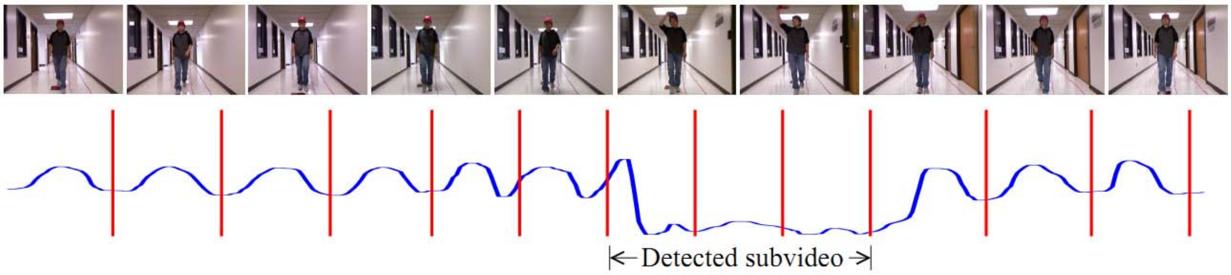

Figure 3. An illustration of period-based video partitioning and the identification of the subvideos of interactive actions. Top: A sequence of video frames. Bottom: Video partitioning shown on one angle signal and the identified subvideos of interactive actions, where red vertical lines indicate the divider frames.

u^* and estimate the period length as before. As shown in Fig. 2(c), the combined frequency response contains much less noise, which simplify the period-length estimation.

B. Period-based Video Partitioning

In this section, we partition the video in terms of the identified period of the non-interactive action. As illustrated in Fig. 3, we expect each video segment after this partitioning to represent one period of the non-interactive action or a short-duration of an interactive action. This way, we can analyze the pose and kinematic features of each video segment to determine whether it describes a non-interactive or an interactive action. One major issue of this video partitioning is that the non-interactive action is not perfectly periodical. For example, when a person walks without interacting with others, his walking speed may still vary from one period to another. In another word, the period-length estimated in the previous section is the most typical one for the considered non-interactive action, but does not mean that all the periods of the non-interactive action bear this same period length perfectly.

In this paper, we address this problem by allowing the period length of the non-interactive action to vary over time. Considering an angle signal $f_i(t), t = 0,1,...,T-1$, we first take its first M consecutive, non-overlapping subsequences $\{f_i(mT_c : ((m+1)T_c - 1))\}$,m = 0, 1, · · · ,M − 1 and then take one most representative subsequence as the typical period of this angle signal. Here the most representative subsequence is defined to be the one with the smallest Euclidean distance to the other M − 1 subsequences. We refer to this most representative length-Tc subsequence as a template Ti. In the following, we partition the angle signal $f_i(t), t = 0,1,...,T-1$ by finding the video segments to match the template T_i. In this video partitioning, we allow the length of each video segment to be different from T_c. In addition, while we construct the template for each angle separately, we finally combine the matching cost from different angles to construct a unified video partitioning.

We first define the matching cost between the template (a length- T_c subsequence) T_i and any subsequence $f_i(t_1 : t_2)$ whose length $(t_2 - t_1)$ may not equal to T_c. By allowing the small change of the period length in the non-interactive action, we uniformly sample T_c points between t_1 and $t_2 - 1$ and then interpolate $f_i(t)$ at these T_c sampled points. We then compute the Euclidean distance between this interpolated length- T_c subsequence and the template T_i as the desired matching cost $C(T_i, f_i(t_1 : t_2))$. By combining the matching cost over all K angles, we define the cost of a video segment $t_1 : t_2$ as

$$w(t_1 : t_2) = \sum_{i=1}^{K} C(T_i, f_i(t_1 : t_2)) \qquad (2)$$

For a complete video partitioning with divider frames 0 < $t_1 < t_2 < \cdots < t_L \le T$, which leads to L consecutive video segments $t_l : t_{l+1}$, $l = 0,1,...,L-1; t_0 = 0$, we define a video-partitioning cost as

$$W(L; t_1, t_2, \cdots, t_L) = \sum_{l=0}^{L-1} w(t_l : t_{l+1}) \qquad (3)$$

subject to the constraints:

$$t_0 = 0 < t_1 < t_2 < \cdots < t_L \le T \qquad (4)$$

$$(1-\varepsilon)T_c \le t_{l+1} - t_l \le (1+\varepsilon)T_c \qquad (5)$$

$$t_L > T - (1-\varepsilon)T_c \qquad (6)$$

where (5) allows the period-length to vary around T_c by a small percentage of ε and (6) drops a possible incomplete period of non-interactive action at the end of the video in the video partitioning. Our goal is to find the optimal L and a set of divider frame indices $0 < t_1 < t_2 < \cdots < t_L \le T$ to minimize the partitioning cost (3), in which the factor $1/L$ is introduced to avoid a bias to produce fewer, longer video segments. In the following, we describe a dynamic-programming approach to solve this optimization problem.

We first compute the upper-bound and lower-bound of L by

$$L_{\max} = \left\lfloor \frac{T}{(1-\varepsilon)T_c} \right\rfloor \qquad (7)$$

$$L_{\min} = \left\lceil \frac{T}{(1+\varepsilon)T_c} \right\rceil \qquad (8)$$

We construct a $T \times L_{\max}$ graph with a 2D table of nodes Ψ(t, l), t = 1, · · · , T ; l = 1, 2, · · · , $L_{\max}$ as shown in Fig. 4, where the node Ψ(t, l) indicates the case of selecting frame t as the l-th divider frame, i.e., t_l = t. We then construct forwarding edges to connect the nodes between two neighboring columns and use them to represent a video segment. For example, an edge from Ψ(t_1, l) to Ψ(t_2, l + 1) indicates that the subsequence $t_1 : t_2$ is taken as the (l + 1)-th video segment in the resulting video partitioning, where l = 1, 2, · · · , $L_{\max}$. We simply use the cost $w(t_1 : t_2)$ defined in (2) as the weight of this edge. We construct the edges only if the corresponding video segment $t_1 : t_2$ satisfy the above three constraints. In this graph, we represent the video segment $0 : t_1$ by introducing an entering node Ψ0 to represent that the first video segment starts from frame $t_0 = 0$. We construct forwarding edge from Ψ0 to the Ψ(t_1, 1) (a subset of nodes in the first column of Ψ), when $t_0 : t_1$ satisfies the above three constraints. Similarly, the edge weight between Ψ0 and Ψ(t_1 , 1) is defined using the cost $w(0 : t_1)$.

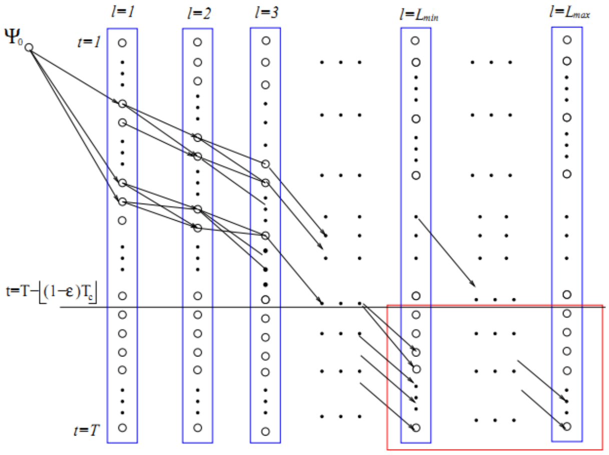

Figure 4. An illustration of the proposed video partitioning algorithm.

In the above table, we can see that partitioning the video at frames $0 < t_1 < t_2 < \cdots < t_L \le T$ can be represented by a path starting from the entering node, traversing nodes Ψ(t_1, l), l = 1, 2, · · · ,L sequentially, and finally ending at Ψ(t_L ,L) in the L-th column of the graph. Using dynamic programming, we can easily find the shortest path (the path with the minimum total weights) between Ψ0 and any node Ψ(t_L ,L) in this graph. This shortest path length, which we denote as W_0 (t_L ,L) (i.e., the minimum total edge weights along the path) equals to $\sum_{l=0}^{L-1} w(t_l : t_{l+1})$, which differs from the desired cost $W(L; t_1, t_2, \cdots t_L)$ only by a factor of $1/L$. In addition, we still need to consider the constraint (6) to make sure the dropped subsequence at the end of the video can not constitute a complete period.

To address this problem, we focus on all the nodes Ψ(t_L , L) that satisfy two conditions:

(C1) $t_L > T - (1-\varepsilon)T_c$, and (C2) $L_{\min} < L < L_{\max}$. These nodes have no outgoing edges in the constructed graph and we refer to them as exiting nodes, as indicated by the nodes in the red box in Fig. 4. We use dynamic programming to compute the shortest path between the entering node Ψ0 and each exiting node Ψ(t_L ,L), and then compute their average path length $\frac{W_0(t_L, L)}{L}$. Finally among all the exiting nodes, we identify the one with the minimum value of $\frac{W_0(t_L, L)}{L}$ and it is easy to verify that the shortest path between the entering node and this identified exiting node describes the desired video partitioning that minimizes the cost (3) subject to the above three constraints. Figure 3 shows the result of video partitioning on an angle signal.

C. Identifying Subvideos of Interactive Actions

In this section, we examine the pose and kinematic features of the video segments obtained in the previous section and classify each of them as either Seg-I which represents an interactive action, or Seg-N which represents the non-interactive action. This process is divided into two steps. First, for each angle i, we check the matching cost $C(T_i, f_i(t_l : t_{l+1}))$ for each video segment. We compute the mean and variance of the L matching costs (for L video segments) for this angle i . If the matching cost $C(T_i, f_i(t_l : t_{l+1}))$ is located two-time standard deviation away from the mean, we add one vote to video segment $t_l : t_{l+1}$ to be classified as Seg-I. We repeat this voting process for all the video segments over all K angles. If the total votes to a video segment is larger than a pre-set threshold T_v, we label this segment by Seg-I. We then temporally smooth the labeling results by using a very simple rule: Given that each video segment has two neighboring video segments (excluding the first and the last ones in the video), we further label a video segment by Seg-I if its two neighbors have been labeled by Seg-I. Finally, we set label Seg-N to all the video segments that are not labeled as Seg-I. After this smoothing, we identify one or more subvideos of interactive actions by aggregating the consecutive video segments that are labeled by Seg-I. While in practice, a person may perform more than one interaction in a video. In this paper, as an exploratory study, we assume in each video the person can only perform interaction no more than once. As a result, we only need to pick one subvideo of interaction action. In our experiment, for each video we simply pick the longest subvideo labeled by Seg-I as the detected interaction action.

IV. Subvideo Matching

In this section, we compare the subvideos identified in the previous section to see whether they describe the same interactive action. We achieve this by defining a matching distance between two subvideos based on their angle signals. To better distinguish the short-duration non-interactive actions, we take the 11 upper-body joint-angle signals as used in the video matching and combine them with the following seven angles and four relative distances:

- Seven additional angles: one angle at head by connecting to two wrists, one angle at hip center by connecting to two wrists, one angle at head by connecting to two ankles, one angle at hip center by connecting to two ankles, one angle at hip center by connecting to the head and the center of two knees, one angle between left forearm (connecting wrist and elbow) and the line connecting hip left and hip right, and one angle between right forearm and the line connecting hip left and hip right.
- Four relative distances: one for the distance between two wrists, two for the distance between the wrist (left and right) and the head, and one for the distance (projected to the depth axis) between the head and the center of two knees. All these four distances are normalized by the distance between the center of two shoulders and hip center.

Denote these 22 signals extracted from the considered two subvideos as $g(t)$, t = 1, 2, · · · , T_g and $h(s)$, s = 1, 2, · · · , T_h, respectively. We define their matching distance as follows:

- For each frame t in the i-th angle signal $g_i(t)$, find the closest matched frame s in $h_i(s)$, with the minimum distance
$$d(g_i(t), h_i(s)) = \sqrt{\sum_{\Delta=-m}^{m} (g_i(t+\Delta) - h_i(s+\Delta))^2} \quad (9)$$
We take the summation of such minimum distances over all the T_g frames and 11 upper-body angles in $g(t)$ and denote it as D_1. In (9), m ⩾ 0 defines a local neighborhood for comparing the two angle signals at given frames.
- Similarly, for each frame s in the i-th angle signal $h_i(s)$, find the closest matched frame t in $g_i(t)$, with the minimum distance as defined in (9).We then take the summation of such minimum distances over all the T_h frames and 11 upper-body angles in $h(s)$ and denote it as D_2.
- We finally define the matching distance between these two subvideos as $D_1 + D_2$.

In practice, we can use this matching distance to decide whether two subvideos describe the same interactive action or not.

IV. Experiments

For the experiment, we collect a set of Kinect videos. In each video, one person performs a non-interactive action and a short-duration of an interactive action, as mentioned above. In our experiments, the non-interactive action is always walking, but the walking speed may be different for different people and vary over time for the same person. We consider 10 different actions that are used for greeting each other by people from different countries, including (A1) bowing; (A2) nodding; (A3) pressing right hand on the left side of the chest; (A4) raising right hand to salute; (A5) waving right hand over head; (A6) waving right hand at a location to the right of the shoulder; (A7) clapping hands; (A8) raising two hands in front of the chest with two thumbs up; (A9) taking off the hat and putting it back on the head again; and (A10) flying kiss. For each person performing each interactive action, we ask him to perform twice and record them into two videos, respectively. Currently, we have five different human subjects and in total we collect 5 × 10 × 2 = 100 Kinect videos. For performance evaluation, we construct ground truth annotations for each video, which include the type of interactive action (one of the above 10) and the starting and ending frames of the interactive action. For all our experiments, we set the threshold T_v = 2 to the voting results as discussed in Section III-C. We set ε = 20% for the allowed period-length variation as discussed in Section III-A. We choose m = 1 for defining the neighborhood in Section IV.

We first evaluate the performance of the proposed algorithm on identifying subvideos of interactive actions. For each video, denote the starting and ending frames of the identified subvideo to be t_1 and t_2, respectively. Let the annotated ground-truth starting and ending frames to be $\hat{t}_1$ and $\hat{t}_2$, respectively. We compute their overlap using the Jaccard coefficient
$$\frac{|(t_1 : t_2) \cap (\hat{t}_1 : \hat{t}_2)|}{|(t_1 : t_2) \cup (\hat{t}_1 : \hat{t}_2)|}$$
where | · | calculates the length of a subsequence. If this coefficient is larger than a preset threshold T_d, we count that the identified subvideo is correct. Table I shows the

TABLE I
THE PERFORMANCE OF THE PROPOSED ALGORITHM ON IDENTIFYING SUBVIDEOS OF INTERACTIVE ACTIONS. THE VALUES OTHER THAN THE RIGHTMOST COLUMN AND THE BOTTOM ROW SHOW THE NUMBER OUT OF THE 10 VIDEOS (2 FOR EACH OF 5 PERSONS) FOR EACH ACTION IN ON WHICH WE CORRECTLY IDENTIFY THE SUBVIDEO OF INTERACTIVE ACTION UNDER DIFFERENT THRESHOLD TD. THE BOTTOM ROW SHOWS THE AVERAGE JARRCARD COEFFICIENTS AND THE RIGHTMOST COLUMN SHOWS THE STATISTICS OVER ALL 100 VIDEOS.

Action	A1	A2	A3	A4	A5	A6	A7	A8	A9	A19	Total
T_d = 0.4	9	7	10	9	10	10	10	10	10	10	95
T_d = 0.5	8	5	10	9	9	10	10	10	9	10	90
T_d = 0.6	6	5	8	7	7	8	10	9	7	8	75
Aver. Jac	0.589	0.456	0.717	0.659	0.687	0.749	0.742	0.720	0.727	0.759	0.681

quantitative results of this experiment. We can see that the proposed algorithm can identify subvideos correctly on 90 out of 100 videos under threshold T_d = 0.5.

We then evaluate the performance of the proposed algorithm on subvideo matching.We first take the ground truth annotated subvideos for testing the subvideo matching. In this paper, we use an evaluation strategy based on the confusion matrix. We know that one subvideo is identified from each video and we have 10 videos in our video dataset that describe each type of interactive action. For each subvideo, we take it as a query and compute its matching distance to all the 100 subvideos (including itself), from which we select the top 10 subvideos with the smallest distance to the query. For each set of the 10 queries that describe the same type of interactive action, we then get 10 × 10 = 100 such top subvideos (a same subvideo may appear more than once in these 100 subvideos). We then divide these 100 subvideos into 10 groups in terms of their ground-truth type of the interactive action. The size of each group is then an element of a confusion matrix as shown in Table II. For example, at the crossing of row "A4" and column "A10", we have a number "5". This indicates that 5 out of the 100 top subvideos resulting from the queries with action type "A4" actually describes the action type "A10". Clearly, the larger the numbers along the diagonal of this table, the better the performance of the subvideo matching, and the maximum possible value for each diagonal element is 100.

Table III shows the confusion matrix of the subvideo matching based on the subvideos identified by the proposed algorithm. Comparing the diagonal numbers of this table with these in Table II which has an average matching rate of 87.5% , we can see that for many types of interactive actions, the use of the subvideos identified by the proposed algorithm performs as well as the use of the ground-truth subvideos identified manually. By taking the average of the diagonal elements, we can get an average matching rate of 82.1% for this experiment. Be reminded again that the proposed subvideo matching developed in this paper is only based on the pose and kinematic features. In practice, we can easily include the temporal coincidence (i.e., the interactive action of two involved persons should occur at the same time) and the spatial coincidence (the persons who are interacting should face each other) to further improve the interaction-detection accuracy.

TABLE II
THE CONFUSION MATRIX OF THE SUBVIDEO MATCHING BASED ON THE GROUND-TRUTH SUBVIDEOS MANUALLY IDENTIFIED FROM THE 100 COLLECTED VIDEOS. NOTE THAT THE MAXIMUM POSSIBLE VALUE FOR EACH DIAGONAL ELEMENT IS 100.

Action	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	**96**	3	0	0	0	0	1	0	0	0
A2	0	**100**	0	0	0	0	0	0	0	0
A3	0	0	**100**	0	0	0	0	0	0	0
A4	0	0	0	**94**	0	5	0	0	0	1
A5	0	0	0	6	**76**	0	0	0	10	8
A6	0	0	0	35	0	**64**	0	0	0	1
A7	1	0	0	0	0	0	**72**	27	0	0
A8	0	0	0	0	0	0	1	**99**	0	0
A9	0	0	0	0	0	4	0	0	**88**	8
A10	0	0	0	0	7	2	1	0	4	**86**

TABLE III
THE CONFUSION MATRIX OF THE SUBVIDEO MATCHING BASED ON 100 SUBVIDEOS IDENTIFIED BY THE PROPOSED ALGORITHM. NOTE THAT THE MAXIMUM POSSIBLE VALUE FOR EACH DIAGONAL ELEMENT IS 100.

Action	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	**83**	16	0	0	0	0	1	0	0	0
A2	18	**82**	0	0	0	0	0	0	0	0
A3	0	0	**96**	3	0	1	0	0	0	0
A4	0	0	0	**82**	0	16	0	0	1	1
A5	0	0	0	6	**69**	0	0	0	26	5
A6	0	0	0	5	0	**80**	0	0	0	15
A7	1	0	0	0	0	0	**80**	20	0	0
A8	4	0	0	0	0	0	7	**89**	0	0
A9	0	0	0	0	0	11	0	0	**82**	7
A10	0	0	0	0	0	1	8	0	13	**78**

VI. CONCLUSIONS

In this paper, we introduced a new approach for detecting human interaction, using Kinect videos as input. Different from many previous works based on action recognition, the proposed approach is unsupervised. We temporally distinguished human actions in a video into a largely periodical non-interactive action and an interactive action. We developed algorithms to estimate the period-length of the non-interactive action, partition the video in terms of the estimated period, and then identify the short-duration subvideo of the interactive action. We finally developed an algorithm to compare the pose and kinematic features of the identified subvideos to seek identical interactive actions, which can be used to decide whether a human interaction occurs. We collected 100 Kinect videos from different people with different interactive actions to test the performance of the proposed algorithms. We found that the proposed algorithms identify subvideos of interactive actions with good overlaps to the ground-truth annotation and the proposed algorithms can produce a 82.1% correct matching rate in the confusion matrix.

REFERENCES

[1] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008, pp. 1–8.

[2] A. Patron, M. Marszalek, I. Reid, and A. Zisserman., "High five: Recognising human interactions in tv shows," in In proceeding of British Machine Vision Conference (BMVC), 2010.

[3] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009, pp. 1593– 1600.

[4] Y. Zhou, S. Yan, and T. S. Huang, "Pair-activity classification by bi-trajectories analysis," in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008, pp. 1–8.

[5] B. Ni, S. Yan, and A. Kassim, "Recognizing human group activities with localized causalities," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 1470–1477.

[6] J. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," ACM Computing Surveys (CSUR), vol. 43, no. 3, p. 16, 2011.

[7] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 1. IEEE, 2005, pp. 166–173.

[8] A. Yilma and M. Shah, "Recognizing human actions in videos acquired by uncalibrated moving cameras," in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 1. IEEE, 2005, pp. 150–157.

[9] H. Zheng, Z. Li, and Y. Fu, "Efficient human action recognition by luminance field trajectory and geometry information," in Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on. IEEE, 2009, pp. 842–845.

[10] P. Natarajan and R. Nevatia, "Coupled hidden semi markov models for activity recognition," in Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on. IEEE, 2007, pp. 10–10.

[11] J. C. Niebles, B. Han, and L. Fei-Fei, "Efficient extraction of human motion volumes by tracking," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 655–662.

[12] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, "Modeling individual and group actions in meetings with layered hmms," IEEE Transactions on Multimedia, vol. 8, no. 3, pp. 509–520, 2006.

[13] D. Damen and D. Hogg, "Recognizing linked events: Searching the space of feasible explanations," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 927–934.

[14] J. Yin and Y. Meng, "Human activity recognition in video using a hierarchical probabilistic latent model," in Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. IEEE, 2010, pp. 15–20.

[15] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis, "Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 2012–2019.

[16] T. Hospedales, J. Li, S. Gong, and T. Xiang, "Identifying rare and subtle behaviors: A weakly supervised joint topic model," vol. 33, no. 12, pp. 2451–2464, 2011.

[17] T. Xiang and S. Gong, "Beyond tracking: Modelling activity and understanding behaviour," vol. 61, no. 1, pp. 21–51, 2006.

[18] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1. IEEE, 2005, pp. 838–845.

[19] T. Xiang and S. Gong, "Video behavior profiling for anomaly detection," vol. 30, no. 5, pp. 893–908, 2008.

[20] Y. Li, C. Xu, J. Liu, and X. Tang, "Detecting irregularity in videos using kernel estimation and kd trees," in Proceedings of the 14th annual ACM international conference on Multimedia. ACM, 2006, pp. 639–642.

[21] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 2. IEEE, 2004, pp. 819–826.

[22] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 935–942.

[23] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 1975–1981.

[24] X. Cui, Q. Liu, M. Gao, and D. N. Metaxas, "Abnormal detection using interaction energy potentials," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011, pp. 3161–3167.

[25] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," Communications of the ACM, vol. 56, no. 1, pp. 116–124, 2013.

[26] J. Rohen, C. Yokochi, and E. Lutjen-Drecoll., "Color atlas of anatomy: A photographic study of the human body," Lippincott Williams & Wilkins, 2006.

Chengzhang Qu received his B.S. degree in applied mathematics from Wuhan University, China in June 2006. He is currently working towards his Ph.D. degree in computer school at Wuhan University, China. His current research interest includes computer vision and machine learning.

Dengyi Zhang received his B.S. degree in department of electronic engineering from Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 2000 and M.S. degree in department of computer science from Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 2003. He is now working as a Professor in Computer School of Wuhan University. His research interests include embedded system design, image processing and pattern recognition.

Yuewei Lin received the BS degree in optical information science and technology from Sichuan University, Chengdu, China, and the ME degree in optical engineering from Chongqing University, Chongqing, China. He is currently working toward the PhD degree in the Department of Computer Science and Engineering at the University of South Carolina. His current research interests include computer vision and image/video processing. He is a student member of the IEEE.

Song Wang received the PhD degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC) in 2002. From 1998 to 2002, he also worked as a research assistant in the Image Formation and Processing Group at the Beckman Institute of UIUC. In 2002, he joined the Department of Computer Science and Engineering at the University of South Carolina, where he is currently a professor. His research interests include computer vision, medical image processing, and machine learning. He is currently serving as the publicity/web portal chair of the Technical Committee on Pattern Analysis and Machine Intelligence of the IEEE Computer Society, and as an associate editor of Pattern Recognition Letters. He is a senior member of the IEEE and a member of the IEEE Computer Society.