Recognition of Composite Human Activities through Context-Free Grammar based Representation

M. S. Ryoo and J. K. Aggarwal
Computer & Vision Research Center / Department of ECE
University of Texas at Austin
{mryoo, aggarwaljk}@mail.utexas.edu

Abstract

This paper describes a general methodology for automated recognition of complex human activities. The methodology uses a context-free grammar (CFG) based representation scheme to represent composite actions and interactions. The CFG-based representation enables us to formally define complex human activities based on simple actions or movements. Human activities are classified into three categories: atomic action, composite action, and interaction. Our system is not only able to represent complex human activities formally, but also able to recognize represented actions and interactions with high accuracy. Image sequences are processed to extract poses and gestures. Based on gestures, the system detects actions and interactions occurring in a sequence of image frames. Our results show that the system is able to represent composite actions and interactions naturally. The system was tested to represent and recognize eight types of interactions: approach, depart, point, shake-hands, hug, punch, kick, and push. The experiments show that the system can recognize sequences of represented composite actions and interactions with a high recognition rate.

1. Introduction

High-level understanding of human activity is essential for various applications, including surveillance systems and human computer interactions. In particular, a human activity recognition system may enable the detection of abnormal activities as opposed to the normal activity of persons using public places like airports and subway stations. Automated human activity recognition may be useful for real-time monitoring of the elderly people, patients, or babies. Several researchers have worked on human activity recognition at various levels [1]. Some researches focus on simple tracking of persons, and others focus on estimating the physical state of persons in the scene. Further, various analyses on human actions have been conducted. Most of the previous researches focused mainly on the recognition of single (i.e. atomic) actions of humans, not on recognition of complex composition of multiple movements or actions [6, 7]. However, recently, understanding semantics of composite actions is getting more and more interest among researchers [3,4,5,9,10].

In this paper, we aim to recognize composite actions and interactions using a context-free grammar (CFG) based representation scheme. Our CFG representation scheme is able to construct a concrete representation for any composite action, and thus enables the system to recognize the defined composite actions based on their representation. Human actions and interactions are usually composed of multiple sub-actions, which themselves are atomic or composite actions. Thus, the representation for composite actions must convey the hierarchical and repetitive nature of the human activities. In addition, the recognition system must be able to recognize represented actions and interactions based on their sub-actions.

Our focus in this paper is at the semantic level, the highest level, of the human activity recognition system. In order to recognize composite actions and interactions, raw pixel-level image sequences must be processed up to semantic descriptions of human activities. We adopt previously developed framework to extract features of body parts from pixel-level images [8]. We discuss how extracted features are applied to estimate poses and gestures of persons. Finally, we present the semantic level representation of general human actions and interactions, and the methodology to recognize represented actions and interactions.

Our recognition framework is composed of several layers: the body-part extraction layer, the pose layer, the gesture layer, and the action and interaction layer. The body-part extraction layer, the lowest layer, estimates numerical status of all body parts for each image frame. Taking those numerical values as parameters, the pose layer extracts poses for each frame. The gesture layer then generates sequences of gestures from given sequences of poses. A pose is the abstraction of the state of one body part, and a gesture is the abstraction of meaningful sub-sequence of those poses. At the highest layer, the action and interactions layer, human activities are represented in terms

of time intervals and the relationships among them. The system detects human activities if there exists a time interval that satisfies all conditions specified in the representation. Various pixel-level techniques are used for the body-part extraction layer. Bayesian networks are used to implement the pose layer, and hidden Markov models (HMMs) are implemented for the gesture layer. At the highest layer, actions and interactions are represented semantically using the context-free grammar (CFG). Following the production rules of CFG, the system is not only able to represent composite actions and interactions naturally, but also able to recognize them.

2. Related works

Park and Aggarwal [6, 7] presented a hierarchical framework to recognize human actions and interactions from pixel level images. The framework abstracts image sequence into poses, gestures, actions, and interactions. Their system uses extracted body part knowledge to estimate poses for each frame, and then estimates one most dominant gesture based on sequence of poses. A gesture recognized through HMMs is directly converted into a single action, represented as the operation triplet. Two operation triplets of different persons are combined to form interactions, and two interactions might be combined to present the cause and effect of interactions. Similarly, Nguyen et al. [4] used hierarchical HMMs in order to recognize two levels of actions.

Ivanov and Bobick [3] presented a hierarchical approach using a stochastic context-free grammar (SCFG) at the highest level. Their work divided recognition into two-levels. At the lower level, HMMs were used to recognize primitive trajectories. The primitive trajectories were treated as terminals for SCFG at the higher level. Their SCFG directly generates the sequence of terminals, i.e. primitives, with defined grammar. Language parsing techniques were used to detect events generated through SCFG. The main disadvantage of this approach was that the user must provide all possible production rules for all possible events, even for large domains. Shi et al. [10] tried to overcome the disadvantage of SCFG through using propagation networks as a representation of actions.

In addition, there is research on event definition and inference in traditional AI fields. Allen and Ferguson [2] presented a definition of temporal intervals, and defined events using interval temporal logic. We adopt their concept on events, defining actions and interactions in terms of time intervals. Furthermore, we explicitly define temporal intervals hierarchically, extending the concept of Allen's event representation.

Nevatia et al. [5] constructed a representation language for general events, following modified Allen's temporal logic. Their representation provided promising results on recognition of composite events. They not only provided

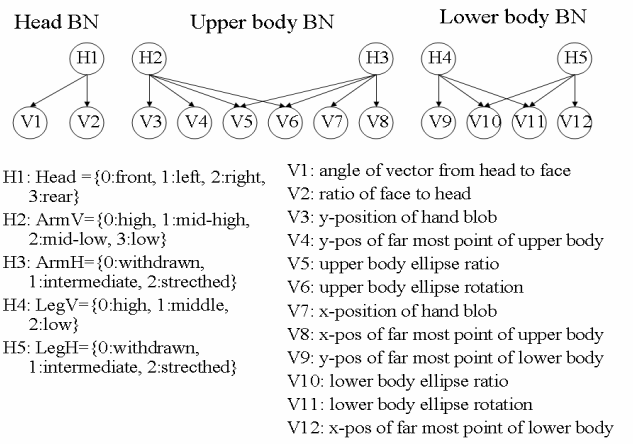

Figure 1: Figure of the Bayesian network and explanation of meaning of nodes. The Bayesian network estimates the state of hidden nodes (i.e. poses), based on observations. The Bayesian network reduces dimensions from twelve into five.

representation scheme, but also illustrated initial results of recognition system using their representation. However, there are two major limitations in their representation language. First, their single-thread composite event corresponds only to a consecutive occurrence of multiple primitive actions. However, it is unlikely in case of single-person human activities, since multiple body parts involve single-person human activities. Secondly, their hierarchy of events is strictly fixed to three levels, and higher-level events can be only composed of lower-level events. This limits constructing high-level composite actions from smaller composite actions, and high-level interactions from smaller interactions.

3. Body-part layer and pose layer

The body-part layer contains pixel-level, blob-level, and object-level processing to extract meaningful information from a sequence of raw images. We use a hierarchical mechanism developed by Park and Aggarwal [8], in order to construct quantitative image features from one input frame. Their system parameterized the state of three body parts (head, upper-body, and lower-body) in terms of ellipses and convex hulls. Maintaining the overall structure of the system, we re-implemented the framework to make it more robust. In addition, our new program explicitly tracks the hand position, extracting additional important features.

In the pose layer, a pose for each body part is estimated based on features extracted by the system's body-part layer. A pose is the abstraction of the body part's static state in one image frame. For each image frame, the pose that best describes instantaneous configuration of the body part is selected based on parameters from the body-part layer. We constructed one-dimensional states for a head pose, describing the torso direction. Upper-body and lower-body

poses have two-dimensional structure, each corresponding to vertical and horizontal positions of a hand and a leg. For example, assume that a person is standing still, facing left with arm fully raised and stretched. Then, his/her head pose will be 'left', upper-body pose will be '<high, stretched>', and lower-body pose will be '<low, withdrawn>'.

Extending Park and Aggarwal's work [6], Bayesian networks are implemented to estimate pose for each body part in each frame. The body-part parameters, estimated from the lower-layer, are converted into discrete values and are treated as observations produced by a specific pose. Bayesian networks estimate the pose for each frame, from the given observation and probabilities in the network. Figure 1 illustrates the structure of the Bayesian network, and possible states for pose of each body part, which is a final output of the pose layer. Note that new features, hand positions, are added. As a result of the pose layer, an input image sequence is converted into a sequence of poses.

4. Gesture layer

A gesture is an elementary movement of a body part. Taking the sequence of poses for each body part as an input, the gesture layer detects possible gestures occurring along the sequence. Essentially, gestures are sub-sequences of whole sequence of poses. The objective of the gesture layer is to recognize a set of all occurring gestures. Each occurring gesture has its starting time and ending time, which might overlap with other gestures.

We construct hidden Markov models (HMMs) to detect the gestures occurring inside the sequence of frames. In order to recognize a sequence of gestures for each body part, we constructed one HMM per gesture. Types of gestures which our system is recognizing in this paper are similar to those in the paper presented by Park and Aggarwal [7]. These include elementary human gestures such as 'arm stretching', 'arm withdrawing', and so on. Each of these HMMs runs in parallel measuring the likelihood of the corresponding gesture based on input. The objective of the gesture layer is to detect which HMM created the sequence of poses and at what point. This is the traditional evaluation problem of the HMM. More specifically, the evaluation problem of the HMM is to determine the probability that a particular sequence of visible states, i.e. poses in our case, was generated by a corresponding model.

Additionally, for each body part, the 'noise HMM' was constructed to cover input sequences that are not related to any gesture we defined. The 'noise HMM' tends to have the highest likelihood for meaningless sequences, making all gestures not to be detected for those sequences.

We use the backward-looking forward algorithm to calculate the likelihood for each HMM. This works same as forward algorithm until detecting the ending point of the gesture. If likelihood of some HMM exceeds the

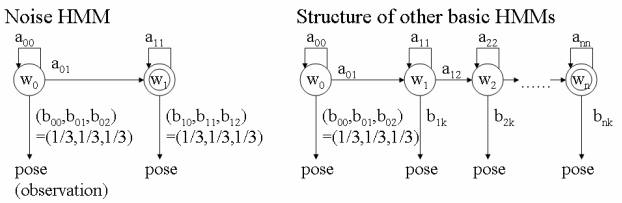

Figure 2: Structure of 'noise HMM' and other HMMs. For each gesture to be recognized, one HMM will be constructed in order to recognize corresponding gestures. Additionally, for each body part, one noise HMM will be created. Probability a_{ij} corresponds to the transition probability from state w_i to w_j. Probability b_{jk} corresponds to the probability of observing k, when the real state of model is w_j.

probability threshold at frame t, we assume that the gesture corresponding to the HMM occurred, and the ending time of that gesture is t. Once the ending time of the gesture is detected, then the algorithm runs a backward algorithm to find the starting point of the gesture. After detecting the starting time and ending time of the gesture correctly, the algorithm proceeds to frame t+1. As a result of the gesture layer, a set of gestures labeled with starting and ending times is created for each body part. Input noises and miscalculation from lower layers are handled in this layer through HMM.

5. Time intervals and predicates

In this section, we discuss the overall structure of the general events, before constructing a specific representation for human actions and interactions. We adopt the concept of interval representation of time presented by Allen and Ferguson [2], in order to construct the representation for general events. We start from associating time intervals with the occurring events. Also, we define temporal, spatial, and logical predicates, which are used to describe relationships among time intervals.

5.1. Structure of the time interval

A time interval intuitively is the time associated with an occurring event. Time intervals we discuss throughout this paper are always associated with designated actions or interactions that we are interested in. In Allen's interval temporal logic [2], a time interval is defined in the linear time line, with a fixed starting point and ending point. They attempted to represent an event by presenting necessary conditions for the event's time interval. Our system follows their approach, but tries to represent hierarchical event explicitly. Since human activities are usually composed of multiple sub-events, the relationships among sub-events' time intervals are the key for the represent of an event.

In Figure 3, relationship among time intervals 'i', 'j', 'k', and 'this' are present to describe the event, 'point' interaction. Each time interval 'i', 'j', and 'k' corresponds

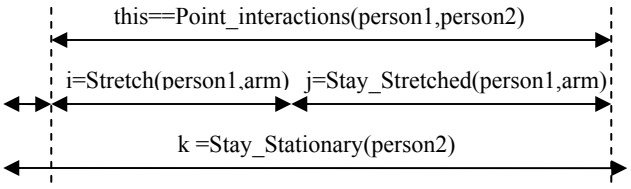

Figure 3: Example of necessary relationship among time intervals for interaction 'point'.

to a smaller event. The variable 'this' is a special variable, always indicating the time interval of the defining action itself. (i) Time interval 'this', assigned for point interaction itself, must start with time interval 'i' and finish with time interval 'j', each assigned for arm stretching event and staying arm stretched event of person1. (ii) Time interval 'i' and 'j' must happen consecutively, and 'i' must be before 'j'. Furthermore, at the same time, (iii) person2 must stay stationary while all these events are occurring.

We should note that time intervals of smaller events are used when representing the relationships. This enables the system to use already defined events to define new higher-level events, providing a concept of hierarchical event representation. We denote all events included in the relationships as 'sub-events' of defining event. That is, event Stretch(person1, arm), Stay_Stretched(person1, arm), and Stay_Stationary(person2) are sub-events of event 'point interaction' in this example.

The concrete definition for the time interval 'this' is the key for hierarchical event representation. Once the time interval 'this' for the corresponding event is correctly represented, the event can be used as a sub-event for other higher-level events.

5.2. Predicates

In the above example, all necessary conditions and relationships for the event are explained in English. We represent those relationships more formally, using three categories of predicates: temporal, spatial, and logical predicates. Temporal and spatial predicates are extremely important when describing human actions and interactions. Temporal predicates express the relationship among time intervals of sub-events. Spatial predicates, on the other hand, describe the relationship between persons involved in the interactions. Logical predicates, '*and*', '*or*', and '*not*', concatenate multiple temporal and spatial predicates to construct overall representation for the event description.

Temporal predicates. Temporal relationships are extremely important when describing human actions and interactions. Usually, actions and interactions of human consist of sequences of sub-events. Temporal predicates not only provide us a mechanism to define such sequential relations, but also help us to provide restricting conditions for the actions and interactions. We directly adopt the temporal relations among time intervals introduced in Allen's interval temporal logic [2]. '*before*', '*meets*', '*overlaps*', '*starts*', '*during*', and '*finishes*' are the predicates defined in Allen's interval temporal logic. Each predicate takes two time intervals as a parameter for the predicates, and decides whether they are true or false. Let a and b be the time intervals, (a_{start}, a_{end}) and (b_{start}, b_{end}).

$before(a, b) \quad <=> a_{end} < b_{start}$
$meets(a, b) \quad <=> a_{end} = b_{start}$
$overlaps(a, b) \quad <=> a_{start} < b_{start} < a_{end}$
$starts(a, b) \quad <=> a_{start} = b_{start}$ and $a_{end} < b_{end}$
$during(a, b) \quad <=> a_{start} > b_{start}$ and $a_{end} < b_{end}$
$finishes(a, b) \quad <=> a_{end} = b_{end}$ and $a_{start} > b_{start}$

Spatial predicates. Spatial predicates define the spatial relationship between two agents or objects. Thus, they can be defined only in terms of interactions. If any interaction contains some spatial predicates, and t is the satisfying time interval of that event, those spatial predicates will always be true in the time interval t.

We designed two spatial predicates: '*near*' and '*touch*'. The '*near*' predicate provides us information on whether two persons are closer than a given relative distance value or not. The distance between two persons is divided by the mean of their heights, producing the relative distance. The '*touch*' predicate is true if and only if the boundary ratio that two persons share is greater than the threshold parameter.

near(person i, person j, threshold) <=>
(Relative distance between i and j) < threshold
touch(person i, person j, threshold) <=>
(Overlapping boundary ratio of i and j) > threshold

Logical predicates. Logical predicates includes '*and*', '*or*', and '*not*' predicate. These are elementary logical predicates. All these predicates can take any relationships as a parameter. The '*and*', '*or*', and '*not*' predicates are defined in an obvious manner. That is, logical predicates can concatenate temporal and spatial predicates to express relationships. The predicate '*and*' holds if and only if relations described in all two parameters are satisfied. The predicate '*or*' holds if more than one of two parameters is satisfied. We say that the '*not*' of a relationship is satisfied if and only if the relationship parameter is false.

6. Atomic actions

Atomic actions are the most elementary component of human activities, which may not be divided into smaller meaningful movements. Atomic components of human actions and interactions are the gestures, recognized through lower-level systems. Therefore, we can construct one atomic action from one gesture. However, gestures solely are insufficient to represent the actions. In order to

represent actions, the system needs to explicitly specify the subject and object of the actions. Following the linguistic theory of 'verb argument structure', we represent atomic actions as <*agent-motion-target*>, following Park's operation triplet [7]. Putting subject and object information together with the gesture, we construct the operation triplet.

For example, 'person 1 stretched his hand to the person 2's head' is an atomic action, because only one gesture is involved in the action. Gesture 'Stretch' is the *motion* of this atomic action. In the operation triplet, 'person 1's hand' is the *agent* and 'person 2's head' is the *target*.

Atomic actions follow the structure of events defined in section 5. When the atomic action is recognized, its corresponding time interval will be equivalent to that of gesture specified in the operation triplet. Since other events cannot affect atomic actions by definition, no other temporal-spatial relationships exist for atomic actions. As a result, the operation triplets are the necessary and sufficient representation of atomic actions.

7. Composite actions

If an action contains two or more atomic actions, it is classified as a composite action. Sub-events of composite actions can be atomic actions, or even other composite actions. The only constraint when constructing composite actions is that only the actions of the same person can become the sub-events. Otherwise, it becomes an interaction, rather than a composite action.

Composite actions follow the structure defined for general events in section 5. The Figure 4 illustrates the time intervals and their relationships for a composite action, 'shake-hands action'. The 'shake-hands action' represents an action that a person is performing in the hand shake interaction. That is, the person stretches one's arm, stays stretched for some period, and then withdraws it. There are three sub-events participating in 'shake-hands action': 'Stretch', 'Stay_Stretched', and 'Withdraw'. Each sub-event has associated variable: 'x', 'y', and 'z'.

7.1. Representation

The representation for composite actions must consist of two parts: a list of variables corresponding to time intervals associated with designated sub-events, and the relationships among those variables. The first component can be represented by associating one symbol name with one sub-event. The second component, which represents necessary conditions for composite actions, is defined through predicates mentioned in section 5. Variables defined and the special variable 'this', representing defining action itself, are used in order to specify the relationships. Therefore, we are able to represent a composite action in terms of the relationship between 'this' and other time interval variables 't1', 't2', ..., which are

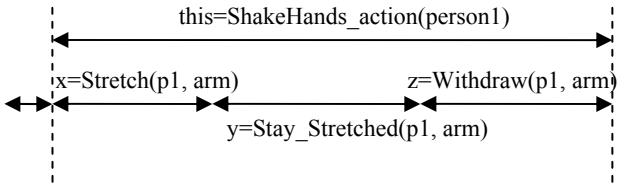

Figure 4: Example illustrating the atomic actions' time intervals and their relationships needed for the composite action, 'shake-hands action'.

satisfying time intervals of sub-events.

As a format of the representation scheme, we use a context-free grammar (CFG). CFG naturally leads the representation to use concepts recursively, enabling the action to be defined based on sub-events. In our representation, atomic actions serve as terminals. On the other hand, composite actions are treated as non-terminals. These non-terminals can be converted to terminals recursively, using production rules.

Our CFG does not generate sequences of poses or gestures directly. Rather, we construct a representation of composite actions using the CFG. A representation built through the CFG describes all participating sub-events, and their relationships. Sub-events can either be atomic actions or other already represented composite actions. Even though the CFG does not create the sequences of poses or gestures directly, we will be able to recognize composite actions through detecting sequences that satisfy the representation constructed with our CFG. With our CFG, we are able to represent any actions if their relationship can be described in terms of the predicates we have defined.

Therefore, the general representation of composite actions can be described using the following context-free-grammar. Non-terminal *Action(i)* indicates action of person i. *Action(i)* can be either an atomic action, or a composite action defined with two components: *ActionDefs(i,var)* and *ActionRelationship(var)*. The first component, *ActionDefs(i,var)*, defines the variables for corresponding time intervals of sub-events. Parameter *var* is defined to be the list of variables associated with sub-events. *ActionDefs(i,var)* is the list of several *def(c,Action(i))*, and this defines the contents of list *var*. Statement *def(c,Action(i))* associates some variable c with the time interval of a denoted sub-event. As a result, list *var* contains a list of variables associated with time intervals of corresponding composing events.

The second component is *ActionRelationship(var)*. With temporal and logical predicates, *ActionRelationship(var)* defines the all necessary conditions for the action using all variables in *var* and special variable 'this'. A combination of any temporal predicates presented in section 5.2 can be used to define *ActionRelationship(var)*. The time interval 'this' satisfying all necessary conditions will be the corresponding time interval for the action.

```
Action(i)
        ->(ActionDefs(i,var), ActionRelationship(var) )
        -> atomic_action(operation triplet)
ActionDefs(i, var)
        -> list( def(c, Action(i)), ActionDefs(i, var-c) )
        -> def(c, Action(i))
ActionRelationship(var)
        -> Logical-Predicate(  ActionRelationship(var),
                               ActionRelationship(var) )
        -> Temporal-Predicate( 'this', var(a) )
        -> Temporal-Predicate( var(a), var(b) )
```

For example, let's look into the composite action 'shake-hands action' again. As we informally defined previously in Figure 4, we associate variable 'x', 'y', and 'z' with sub-events 'Stretch', 'Stay_Stretched', and 'Withdraw'. Then, relationships are represented in terms of predicates: *meets(x, y)*, *meets(y, z)*, *starts(x, this)*, and *finishes(z, this)*. Therefore, formal representation of 'shake-hands action' is defined through our CFG scheme as follows.

```
Stretch_hand(i) =
   atomic_action(<person i's hand, stretch, other person>)
Stay_Stretched_hand(i) = atomic_action
   (<person i's hand, stay stretched, other person's hand>)
Withdraw_hand(i) =
   atomic_action(<person i's hand, withdraw, null>)
SHActionDefs(i, var) = list(
      def('x', Stretch_hand(i)),
         list(
            def('y', Stay_Stretched_hand(i)),
            def('z', Withdraw_hand(i)) )
      )
SHActionRelationship(var) =
      and( meets('x', 'y'),
         and(
            meets('y', 'z'),
            and(starts('x', 'this'), finishes('z', 'this'))
         )
      )
ShakeHands_action(i) =
   (SHActionDefs(i, var), SHActionRelationship(var) )
```

8. Interactions

Interactions are composed of the actions and/or interactions of two persons. In the case of actions, actions were classified into atomic actions and composite actions. However, all interactions have composite characteristics. Therefore, except for the fact that sub-events of interactions can be actions of both persons, the CFG production rule, i.e. representation scheme, of interactions is almost identical to that of composite actions. Further, spatial predicates also can be used to describe relationships for interactions.

```
Interaction(i, j) ->
(InteractionDefs(i, j, var), InteractionRelationship(i, j, var))
InteractionDefs(i, j, var)
   -> list( def(c, Interaction(i, j)),
               InteractionDefs(i, j, var-c) )
   -> list( def(c, Action(i or j)), InteractionDefs(i, j, var-c) )
   -> def(c, Action(i or j))
   -> null
InteractionRelationship(i, j, var)
   -> Logical-Predicate(  InteractionRelationship(i, j, var),
                          InteractionRelationship(i, j, var))
   -> Temporal-Predicate( 'this', var(a) )
   -> Temporal-Predicate( var(a), var(b) )
   -> Spatial-Predicate(person i, person j, threshold)
```

The following example shows how a 'hand-shake' interaction can be represented by following our CFG scheme. Already defined composite actions, 'shake-hands action' of two persons, are used as sub-events of the interaction 'shake-hands interaction'. If person i and j do the action 'shake-hands action' concurrently, and their hands touch, we regard it as a hand shake interaction.

```
Touching_interaction(i, j) = ( null, touch(i, j, 0) )
ShakeHandsDef(i, j, var) =     list(
        def('x', ShakeHands_action(i)),
        list(  def('y', ShakeHands_action(j)),
                  def('z', Touching_interaction(i, j))
        )
     )
ShakeHandsRelationship(i, j, var) =
     and(
        and( during('z', 'x'), during('z', 'y') ),
        and(
           starts('x', 'this'),
           finishes('x', 'this') )
     )
ShakeHands_interactions(i, j) =
   (ShakeHandsDef(i,j,var), ShakeHandsRelationship(i,j))
```

9. Recognition

Detecting time intervals in which an action or interaction occurred is significant part of the recognition. If an action's time interval satisfies all temporal relationships specified in the representation, and participating persons satisfies all spatial relationships in that time interval, then we conclude that the action or interaction is recognized. Time intervals for atomic action can be directly detected through finding time intervals of the gesture specified in operation triplet. For composite actions, an occurring time interval can be detected through finding time intervals that satisfy all temporal relationships needed for variable 'this'. In case of interactions, spatial relationships between two persons also need to be satisfied in the time interval.

9.1. Atomic actions

An atomic action is represented in terms of operation triplet. By definition, an occurring time interval of an atomic action is that of a gesture specified through the *motion* term in operation triplet <*agent, motion, target*>. If the gesture layer recognized a gesture specified in *motion* term of the triplet, and its subject and object corresponds to the *agent* and *target* term of operation triplet, the system concludes that the atomic action is recognized in that time interval.

9.2. Composite actions and interactions

Representation of composite actions and interactions has two components: variable definition and their relationships. In order to recognize a composite action or an interaction, the system first detects all possible time intervals for each variable in a variable list. Since each action or each interaction can occur multiple times, each variable can correspond to multiple time intervals. Finding time intervals for each variable is equivalent to recognizing the corresponding action or interaction for that variable.

Once the time intervals for variables in a variable list are found, the system needs to check whether any combination of the time intervals satisfies all relationships. If there are n variables and $m_1, m_2, \ldots, m_n$ number of time intervals for each variables, then there exist $\Pi_{i=0 \text{ to } n} m_i$ possible combinations of (variable, time interval) pairs. This is a traditional constraint satisfaction problem. The system must find a specific combination of (variable, time interval) pairs that satisfies relationships, among all possible combinations.

Since our representation for actions and interactions has a hierarchical structure, i.e. one action or interaction has multiple sub-events, our action and interaction recognition is done in hierarchical way. If a composite action or an interaction A has action B and C in its variable list, i.e. B and C are sub-events of A, then the recognition system first recognize action B and C. If B and C are composites themselves, they again trigger recognition of their sub-events in the variable list. At some point, all the sub-events will be atomic actions, which the system recognizes using the algorithm described in 9.1. This is similar to tree traversal where actions and interactions are nodes, variable lists specify edges, and atomic actions are leaves. In order to recognize the root action or interaction, the system must recognize its child. This process continues until the system reaches the leaves. Once the system reaches leaves, the system is able to compute time intervals of composite actions or interactions that have atomic actions as sub-events. The system traverses back to the root, recognizing all internal nodes from leaves to the root.

The constraint satisfaction problem is a NP-hard problem, which requires $O(\Pi_{i=0 \text{ to } n} m_i * t^2)$ time complexity

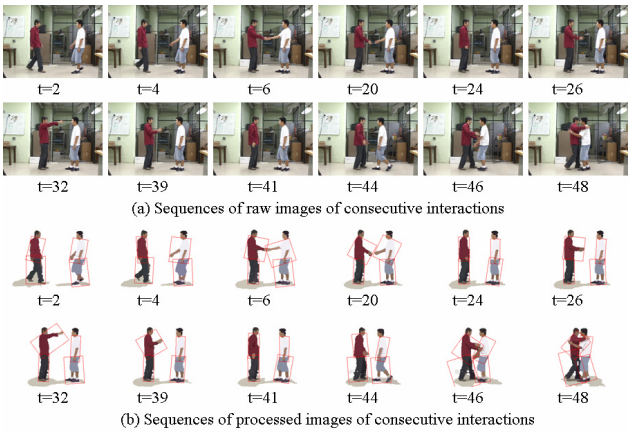

Figure 5: Fig (a) shows sequences of raw images of consecutive three interactions: shake hands, point, and hug. Fig (b) illustrates processed sequence of images by body-part layer.

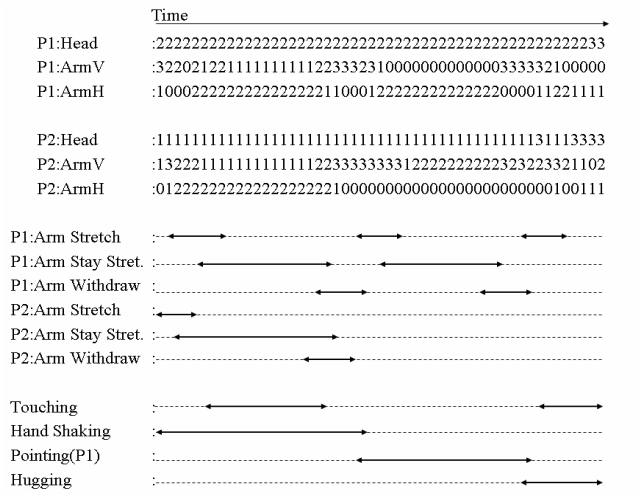

Figure 6: Outputs of the pose layer, the gesture layer, and the actions and interaction layer. Time intervals of atomic actions and interactions are presented.

in our case, where t is the total number of frames. Focusing on linear characteristics of actions and interactions, and forcing additional constraints on relationships and time intervals, the time complexity can be reduced to $O(t * r)$, where r is the number of relationships per action.

10. Experimental results

We recognized the following eight two-person interactions through our system: approach, depart, point, shake-hands, hug, punch, kick, and push. Interaction videos taken by Sony Handy Cam were converted into sequences of image frames with 320*240 pixel resolution, obtained at a rate of 15 frames per sec. Six pairs of persons participated in the experiment and 24 sequences were obtained. In each sequence, participants were asked to

perform a number of above interactions consecutively and continuously. Overall, each interaction was performed 12 times total throughout all sequences.

The representations for the eight interactions were constructed manually using our CFG-based representation scheme. Usually, a composite action is first defined in order to represent meaningful one-person movement in the interaction. For example, in the previous sections, the composite action 'shake-hands action' was defined first in order to represent interaction 'shake-hands interaction'. The composite action 'shake-hands action' and the interaction 'touching' were sub-events.

Figure 5 and 6 show the intermediate outputs of each layer. In this experiment, two persons performed three interactions consecutively: shake-hands, point, and hug. The body-part layer extracts features for each body parts per frame. Figure 5 shows the sequences of raw images, and processed images for extracting body-part parameters. Once the features for each frame are extracted, the pose layer converts them into discrete pose for each body part. The gesture layer converts sequences of poses into sequences of gestures. The recognition algorithm provided in section 9 is then used to recognize interactions based on information from the gesture layer. Figure 6 shows the result of the pose layer, the gesture layer, and the final result of interaction recognitions.

Table 1 shows the performance of our recognition system. Because of the accurate representation on composite actions, the system is superior to all previous systems. Moreover, the results are obtained from sequences of consecutive interactions, not segmented manually. The system was able to recognize sequences of actions and interactions with high degree of accuracy.

11. Conclusion and future works

We presented the general methodology for automated recognition of complex human actions and interactions. The fundamental idea is to use the CFG-based representation scheme to represent composite actions and interactions. The CFG-based representation scheme provides a formal method to define occurring time intervals of composite actions and interactions. The idea of representing complex actions and interactions as a composition of simpler actions and interactions was the key. Our experiments show that the system can represent and recognize composite actions and interactions with high recognition rate.

The novelty of our work is on the framework to represent and recognize high-level hierarchical actions from raw image sequence. Our representation explicitly captures the hierarchical nature of actions and interactions. Our system has the ability to use represented actions as sub-events of higher-level actions, thereby minimizing the redundancy.

The potential of our work is that our system is able to recognize even higher-level composite actions and interactions. Our system can recognize any actions and interactions if their time intervals can be defined properly through our CFG-based representation scheme. Our framework is also able to handle noisy inputs through HMMs. However, current framework cannot process large scale errors, such as insertion or deletion of sub-events. In the future, we plan to take probabilistic nature of actions into consideration. Also, we aim to develop methodology for our system to learn activity representations based on large training sets.

interaction	total	correct	accuracy
approach	12	12	1.000
depart	12	12	1.000
point	12	11	0.917
shake hands	12	11	0.917
hug	12	10	0.833
punch	12	11	0.917
kick	12	10	0.833
push	12	11	0.917
total	96	88	0.917

Table 1: Recognition accuracy of the system

References

[1] J. K. Aggarwal and Q. Cai, Human Motion Analysis: A Review, CVIU 73(3), pp. 295-304, 1999

[2] J. F. Allen and G. Ferguson, Actions and Events in Interval Temporal Logic, Journal of Logic and Computation, 4(5):531-579, 1994.

[3] Y. A. Ivanov and A. F. Bobick, Recognition of Visual Activities and Interactions by Stochastic Parsing, IEEE Transactions on PAMI no. 8, pp. 852-872, August 2000.

[4] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui. Learning and Detecting Activities from Movement Trajectories Using the Hierarchical Hidden Markov Models, pp. 955-960, CVPR'05 - Vol 2, 2005.

[5] R. Nevatia, T. Zhao, and S. Hongeng, Hierarchical Language-based Representation of Events in Video Streams, Proceedings of the Workshop on Event Mining, 2003.

[6] S. Park and J. K. Aggarwal, A Hierarchical Bayesian Network for Event Recognition of Human Actions and Interactions, ACM Journal of Multimedia Systems, special issue on Video Surveillance, 10(2), pp. 164-179, 2004

[7] S. Park and J. K. Aggarwal, Semantic-level Understanding of Human Actions and Interactions using Event Hierarchy, IEEE Workshop on Articulated and Nonrigid Motion, 2004.

[8] S. Park and J. K. Aggarwal, Simultaneous tracking of multiple body parts of interacting persons, CVIU 102(1), pp. 1-21, April 2006.

[9] C. Pinhanez, Representation and Recognition of Action in Interactive Spaces, Ph.D thesis, MIT media lab, 1999.

[10] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation Networks for Recognition of Partially Ordered Sequential Action, pp. 862-869, CVPR'04 – Vol 2, 2004.