Hierarchical Recognition of Human Activities Interacting with Objects

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

Abstract

The paper presents a system that recognizes humans interacting with objects. We delineate a new framework that integrates object recognition, motion estimation, and semantic-level recognition for the reliable recognition of hierarchical human-object interactions. The framework is designed to integrate recognition decisions made by each component, and to probabilistically compensate for the failure of the components with the use of the decisions made by the other components. As a result, human-object interactions in an airport-like environment, such as 'a person carrying a baggage', 'a person leaving his/her baggage', or 'a person snatching another's baggage', are recognized. The experimental results show that not only the performance of the final activity recognition is superior to that of previous approaches, but also the accuracy of the object recognition and the motion estimation increases using feedback from the semantic layer. Several real examples illustrate the superior performance in recognition and semantic description of occurring events.

1. Introduction

Surveillance cameras are becoming popular these days. Increased availability of CCTVs and other monitoring equipments in public places has led to increased demand of automated high-level surveillance systems. Human activities that these types of systems would like to recognize are not the simple gestures or actions of a single person. Rather, the goal of these systems is to analyze and document complicated ongoing human activities where humans and several objects drawn from multiple categories participate in activities. Most human activities in public places involve objects, and thus a system for the recognition of high-level human-object interactions is essential for constructing automated surveillance systems, smart spaces, and human-computer interaction systems in public. For example, in the case of the surveillance system for an airport environment, a system needs to consider object information as well as its movements to analyze and distinguish 'a person simply carrying his/her suitcase' from 'a person snatching another's suitcase in the absence of the person'. In this paper, we present a novel framework for the recognition of human-object interactions composed of 3 main components: the component for object recognition, motion estimation, and semantic-level activity recognition.

Several prior researchers have considered recognition of hierarchical human-object interactions. However, a system that integrates the object recognition, the motion estimation, and the semantic-level analysis for the reliable recognition of hierarchical human-object interactions has not been studied in depth previously. Previous syntactic approaches [2,5] were able to recognize human activities with objects, but they were limited on recognizing semantically complicated activities with concurrent sub-events. Nevatia *et al.* [6] also presented a system to recognize humans interacting with objects. Their system was able to recognize human-object interactions with three levels of hierarchy, but the overall recognition process was strictly dependent on the success of its object recognitions. Ryoo and Aggarwal [8]'s system was general enough to recognize continued human activities with any levels of hierarchy, but did not attempt to recognize human activities with objects. On the other hands, Moore *et al.* [3] constructed the system that compensates for the failures of the object classification with the recognition results of simple actions. Even though the actions their system recognized were simple actions of a single person, their system was able to cope with failures of the object recognition or the action recognition component.

The recognition framework proposed in this paper addresses three key issues, adopting the advantages of previous systems and improving the drawbacks of them. First of all, as mentioned above, the system must be able to recognize human activities with several objects drawn from multiple categories. This suggests that the system needs to consider the object classification problem as well as the recognition of object motion. Secondly, the system must aim to recognize high-level activities, which are usually hierarchical. Finally, the system must recognize human-object interactions reliably and correctly even when one of its components, object recognition for example, fails. Constructing a system that achieves three goals simultaneously is a challenging problem.

We have constructed a probabilistic framework where the three main components (object recognition, motion estimation, and semantic-level activity recognition) complement each other to handle noise and the uncertainties of inputs. Interactions among components play a key role in achieving above-mentioned three goals. We focus on the fact that an object has its own functionalities and thus different types of human activities involve different types of objects. Detected objects and motions enable the semantic layer to recognize high-level activities, while the semantic-level analysis of an activity may help the object layer or the motion layer to recover from failure by providing feedback.

Our focus in this paper is on the semantic layer, which recognizes high-level human-object interactions. We present a reliable recognition algorithm that is able to cope with object recognition or motion estimation errors. An algorithm for detecting the time interval of occurring activities and calculating the probability associated with that interval has been developed. The ability to cope with errors not only increases the recognition performance, but also enables the semantic layer to provide feedback to the other layers. For example, if the system recognized an activity 'person carrying his/her suitcase', then the object that participated in that activity must be a suitcase.

We introduce the overall framework in section 2 with detailed explanations of the segmentation layer, the object layer and the motion layer. Section 3 describes formal language-like representation that our semantic layer uses to recognize human-object interactions. Actual recognition algorithm is presented in section 4, where we present mechanism to associate probability with time intervals to cope with erroneous inputs. Experimental results are shown in section 5. The system recognizes high-level human-object interactions occurring in an airport-like environment. The conclusions are stated in section 6.

2. Framework

For the reliable recognition of human-object interactions, we designed a framework composed of four layers: the segmentation layer, the object layer, the motion layer, and the semantic layer. Each of these four layers has its own functionalities. The role of the segmentation layer is to segment and track objects in the scene using pixel-level and blob-level processing. The object layer identifies categories of segmented objects, while the motion layer estimates movements of objects. The results of the object layer and the motion layer are given to the semantic layer, which takes advantage of detected objects and their motion in order to recognize the final high-level activity. In the semantic layer, the activities are recognized hierarchically from most simple activities, i.e. atomic actions, to composite human-object interactions.

Our semantic layer represents a high-level activity in terms of its sub-events using a language-like representation scheme, and probabilistically recognizes the represented activity with a hierarchical matching algorithm. Our semantic layer has two significant advantages over traditional statistical methods such as dynamic Bayesian networks (DBNs) that has been commonly used for the activity recognition [7]. First, our semantic layer is able to deal with high-level activities composed of sub-events with various types of temporal relationships. DBNs are able to model activities with sequential sub-events, but they lack the ability to model sub-events with concurrent relationships such as 'sub-event1 must occur *during* sub-event2' or 'sub-event1 and sub-event2 must occur exactly *at the same time*'. Secondly, our semantic layer requires significantly less amount of training data to learn high-level activities, since the system incorporates expert knowledge on temporal structure of activities instead of learning them solely from the training data.

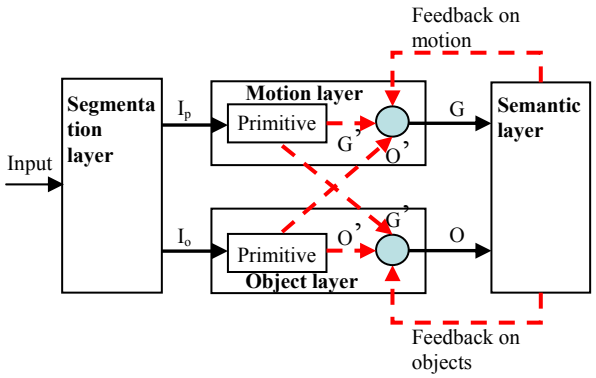


Figure 1: Details of the framework for recognition of high-level human-object interactions.

Furthermore, unlike most previous systems, the layers in our framework are designed to influence each other. The object layer classifies objects not only based on features extracted in the segmentation layer, but also by the decision made by the motion layer and the semantic layer. Similarly, the motion layer estimates movements of objects using information from the segmentation layer, the object layer, and the semantic layer. High-level activities are recognized in the semantic layer, using the outputs of the object recognition and the motion estimation. A failure of recognition in the object layer or the motion layer does not imply the failure of recognition of the semantic layer. As a consequence, the semantic layer is able to help the other layers via feedback when they fail in recognition. The overall framework of our system is shown in the figure 1.

2.1. Segmentation layer

Algorithms for background subtraction, blob detection, and blob tracking are used for our segmentation layer. Our segmentation layer basically segments a cluster of

separated blobs as one object. However, when a person is carrying objects such as suitcases or boxes, object blobs and human blobs form one large cluster rather than two separated clusters. In order to segment objects such as suitcases or boxes from the person who is carrying it, a blob-based version of Haritaoglu *et al.*'s algorithm [4] is used. Their algorithm uses symmetry and periodicity information of people to segment objects from people.

As a result of the segmentation layer, the system estimates positions, shapes, and movements of the objects in the scene.

2.2. Object layer and motion layer

The object layer and the motion layer are designed to take advantage of decisions made by each other. Our intention is to make the recognition process of objects and motions more reliable by considering the relationship between objects and their motions. However, this design principle generates a cycle in the process. We have to know the output of the object layer in order to recognize the motions. The recognition process of objects needs outputs of the motion layer.

The system constructed by Moore *et al.* [3] avoids this cycle between the object layer and the motion layer by giving a priority to the decision of the object layer to that of the motion layer: In most situations, objects are decided first and motions are estimated based on the recognized object. Only when the object layer failed to identify an object, the motion layer is able to help the object layer. There is no cycle in this process, since the system always tries to recognize the object first. The system assumes that the object layer either correctly recognizes an object or labels it as an unclear object.

Our system is designed to overcome this cycle of process by constructing a primitive object module and motion module inside the object layer and the motion layer. The primitive modules are basic classifiers that make a decision solely based on visual observations, without outputs from each other. The two primitive modules are independent. The object layer and the motion layer avoid the cycle of process by treating the primitive module of each other as an estimation of decision of each other. Any of the previously developed object recognition and motions estimation techniques can be safely adopted for each module.

Primitive object module: We use a k-nearest neighbor (k-NN) classifier to recognize objects. The classifier uses six features that have been used commonly for the object classification. Area, height, width, angle of major axis, compactness, and mean color are the features used to classify objects.

Primitive motion module: A hidden Markov model (HMM) is constructed for each motion. HMMs have been widely used for gesture recognition [9]. A HMM treats features extracted from each object, such as 'change of the center of mass' in our case, as 'observations' generated by the hidden nodes of the model. Motion is detected at local maxima of the probability of a HMM generated current sequence of observations. A forward algorithm of HMM is used to detect the ending time of a motion, and a backward algorithm is used to detect the starting time.

A naïve Bayesian classifier is constructed for each layer to make the final decision. The object layer and the motion layer use each other's primitive module to take advantage of each other. The motion layer estimates object motions based on its primitive module, the primitive object module and, feedback from the semantic layer. The object layer classifies objects using its primitive module, feedback from the semantic layer, and the entire history of motions estimated by the primitive motion module. Dotted line of figure 1 illustrates this process.

3. Semantic layer: representation

In order to recognize high-level human-object interactions, the system must have knowledge on the temporal and spatial structure of the human activities that it desires to recognize. Our approach is to make human's conceptual knowledge on human-object interactions encoded as an activity representation, whose format is similar to that of a programming language.

We extend Ryoo and Aggarwal's representation [8] to represent human-object interactions formally. Their representation scheme describes a high-level activity based on its sub-events and their temporal [1], spatial, and logical relationship. Unlike previous approaches [2,5,7], their representation scheme is able to represent human activities composed of sequential and concurrent sub-events. If an activity has no sub-event, they call it an 'atomic action'. If not, they call it 'composite activity'. In principle, sub-events of an activity can be any composite activities or atomic actions that have been represented, suggesting that the activities are represented hierarchically.

The major extension made in our activity representation is on the syntax to describe participating objects of interactions. Similar to their previous work, an atomic action is represented in terms of the 'operation triplet' : *<agent, motion, target>*. The difference is that now the agent and the target is not limited to a person; it can be an object from a number of categories. In addition, all participating objects must be specified when representing composite human activities with objects. The extended version of CFG-based representation scheme is as follows:

InteractionName(ParameterObjectParticipants) = {
InteractionDefs,
InteractionRelationship
};

InteractionDefs specifies the list of sub-events and *InteractionsRelationship* specifies necessary relationship needed among sub-events. *ParameterObjectParticipants* are list of all participating objects, which can be described as *ObjectClass ObjectName*. For example, the interaction 'person stealing another's suitcase' is as follows:

```
Steal(Person p1, Suitcase s1, Person p2) = {
   list( def('i', Carry(p1, s1)),
         list( def('j', Stay(s1)), def('k', Carry(p2, s1)) ) ),
   and( and( equals('this', 'k'),
         and(meets('i', 'j'), meets('j', 'k')) )
};
Carry(Person p1, Suitcase s1) = {
   list( list(def('mlp', MoveL(p1)), def('mls', MoveL(s1))),
         list(def('mrp', MoveR(p1)), def('mrs', MoveR(s1))) ),
   and( touching(p1, s1),
         or( and( equals('this', 'mlp'), equals('mlp', 'mls') ),
             and(equals('this', 'mrp'), equals('mrp', 'mrs') )))
};
```

4. Semantic layer: recognition

In the semantic layer, high-level human-object interactions are recognized using our new probabilistic algorithm. The recognition of human-object interactions is done based on the matching between the language-like representation of the activities constructed by human users and the recognition results from the object layer and the motion layer. Given the detection results of objects and motions, the semantic layer of the system must detect the valid combination of objects and motions that matches the representation of activities with high probability.

4.1. Time interval detection algorithm

A 'time interval' is time associated with an occurring activity, composed of starting time and ending time. 'Participants' are all objects which are involved in the activity. Therefore, recognizing an activity is equivalent to detecting (participants, time interval) pairs that satisfy the representation of the activity with high probability. In this sub-section, we present an algorithm which searches for combinations of objects and motions in order to detect valid participants and time intervals of represented activities. We show that time intervals can be computed based on the object and motion detection results. The algorithm presented in this sub-section calculates possible candidate (participants, time interval) pairs of activities based on objects and motions detections, while the probability (or confidence) of them are computed in the sub-section 4.2.

The hierarchy tree of the activity illustrates the process of our algorithm. The structure of the activity described by its language-like representation is interpreted into the hierarchy tree. A node of the hierarchy tree corresponds to an activity, and an edge specifies which activity is the sub-event of which activity. If an activity is a sub-event of another activity, the former becomes a child of the latter in the hierarchy tree. By definition, all leaf nodes are atomic actions, while all internal nodes are composite activities. Particular temporal relationships exist among siblings of the tree, and spatial relationships exist among objects associated with siblings. Figure 2 shows the hierarchy tree of the interaction 'steal (p1, s1, p2)'.

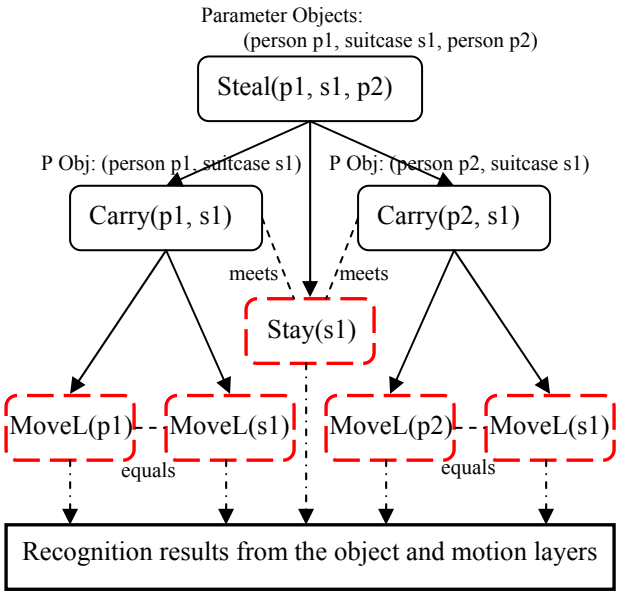


Figure 2: Example hierarchy tree of the interaction 'steal (p1, s1, p2)'. Dotted boxes are atomic actions. The activity 'carry' actually has two more sub-events that correspond to 'move right' actions of a person and a suitcase, which has been omitted here.

Our algorithm to recognize human activities essentially is a hierarchical matching algorithm using the hierarchy tree. At each node, the system matches the temporal structure of the activity with time interval detection results of the sub-events, and checks whether the participant objects of sub-events satisfy the spatial structure of the activity or not. Given a combination of (participants, time interval) pairs where each of them are associated with a child node, the system checks whether the assignments on child nodes satisfy necessary spatio-temporal relationship among them. For each valid combination of (participants, time interval) assignments on child nodes, the time interval of the parent node is computed by calculating the range of the special time interval 'this' associated with itself in the representation. Participants of the parent node can be calculated based on the participants of sub-events. Each participant of the parent node corresponds to one (or more) of the participant of sub-events. Figure 3 shows the pseudo code of the algorithm.

Searching for valid combinations of a (participants, time interval) pair assignments is a typical constraint satisfaction problem. There are multiple candidate assignments for each

child node, and the system must find a valid combination of them which satisfies spatio-temporal constraints. For efficiency of our algorithm, we take advantage of the linear characteristics of the activities. We limit our representation to not use two identical activities with identical objects as sub-events of one activity. With this constraint, most of the activities show linear characteristic; among time intervals of sub-events associated with identical participants, only the most recent one is involved in the activity.

The overall process is done bottom-up. Initially, leaf nodes (i.e. atomic actions) are recognized through searching for the recognition results of objects and their movements. Thus, 'participants' associated with an atomic action is identical to a single detected object, and 'time intervals' associated with an atomic action is identical to those of the detected object's motions. Once (participants, time interval) pairs of atomic actions are recognized, other high-level activities constructed based on the atomic actions can be recognized bottom-up. At each internal node, matching is performed between each valid combination of (participants, time interval) assignments on child nodes and the spatio-temporal structure needed among them.

When detecting (participants, time interval) pairs for atomic actions, we do not discard the time intervals of object motions even when the object label mismatches the operation triplet with high probability. The penalty for the mismatch will be covered when calculating the probability of occurrence of the atomic action in that time interval. The probability of occurrence of the atomic action will be low if the object detection result does not match the operation triplet of the atomic action. The main idea is to let the probability calculation mechanism decide the mismatch between the representation of the activity and the recognition results from the lower layers. The role of the time interval detection algorithm is to provide as many valid candidates of time intervals associated with the activity as possible for the probability calculation system to take advantage of them.

4.2. Calculating the probability of occurring activities

The objective of the algorithm presented in this sub-section is to calculate a probability of an occurring activity associated with a time interval, given the sequence of images. If we denote images from frame 1 to T as I^T, then the conditional probability of the activity R occurred in the time interval $<s, e>$ can be expressed as $P(R^{<s, e>} \mid I^T)$. The goal of our algorithm is to calculate $P(R^{<s, e>} \mid I^T)$ based on the recognition results of objects, $P(O_i=j \mid I^T)$ where O_i is the id of the object and j specifies the category, and the recognition results of motions, $P(M_i^{<s, e>}=j \mid I^T)$ where M_i is the id of the motion and j specifies its category.

In order to calculate probability of a high-level activity,

```
RECOGNIZE(Activity a) {
  for all sub-events s_i, list[i] = RECOGNIZE(s_i);
  for each combination c = (j_1, ..., j_n) where j_i ∈ list[i]
    if (CheckTemporal(c.t)==false) continue;
    else result_t = CalculateThis(c.t);
    for each c.o[i] that is a defined object of a
      let all.o[i]^k be a object of sub-event that has to
        be identical to c.o[i].
      if(o[i]^1=o[i]^2=...=o[i]^n) result_o[i] = o[i]^1;
      else continue;
    result.add((result_o, result_t));
  return result;
}
```

Figure 3: Pseudo code of the hierarchical recognition algorithm,

we use the dependency information between the activity and its sub-events. The hierarchy tree illustrates dependencies among the activities similar to the Bayesian network. Activities associated with child nodes depend on the activity associated with a parent node. By the definition of the operation triplets, the object and its motion specified in an operation triplet depend on that atomic action, i.e. the leaf node. The main difference between the dependency among nodes in the hierarchy tree and those in the Bayesian network is that siblings of the hierarchy tree are not conditionally independent given the parent node; sub-events tend to occur together, implying that they are highly correlated.

We denote a union of sub-events of each element in set S as $Sub(S)$. When an element *'a'* of set S does not have any sub-events, the $Sub(S)$ is defined to be $Sub(S\text{-}a) \cup$ *'a'*. Then, the probability $P(R^{<s, e>} \mid I^T)$ can be enumerated using the dependency among nodes, as follows:

$$
\begin{aligned}
P\left(R^{\langle s,e\rangle} \mid I^T\right) &= P\left(\{R\} \mid sub\left(\{R\}\right)\right) * \\
&\quad P\left(sub\left(\{R\}\right) \mid sub\left(sub\left(\{R\}\right)\right)\right) * \ldots * P\left(sub^d\left(\{R\}\right) \mid I^T\right) \\
&= \prod_{i=0}^{d-1} P\left(sub^i\left(\{R\}\right) \mid sub^{i+1}\left(\{R\}\right)\right) * P\left(sub^d\left(\{R\}\right) \mid I^T\right) \\
&= \prod_{i=0}^{d-1} P\left(sub^i\left(\{R\}\right) \mid sub^{i+1}\left(\{R\}\right)\right) * P\left(a_1, \ldots, a_n \mid I^T\right)
\end{aligned}
$$

where *a1, a2, ..., an* are leaf nodes of the tree and d is the depth of the tree.

Because of the characteristics of our representation, we can safely assume that an activity occurs if and only if all of its sub-events occur. That is, for all set of siblings S in the tree, $P(S \mid sub(S)) = 1$

Therefore, the $P(R^{<s, e>} \mid I^T)$ can be simplified into the product of conditional probabilities among atomic actions, objects, and motions. If we assume conditional independency among recognitions made by the object layer and the motion layer, the probability $P(R^{<s, e>} \mid I^T)$ is enumerated as follows:

$$
\begin{aligned}
&P\left(R^{\langle s,e\rangle}|I^T\right)\\
&=\prod_{i=0}^{d-1}P\left(sub^i\left(\{R\}\right)|sub^{i+1}\left(\{R\}\right)\right)*P\left(a_1,\ldots,a_n|I^T\right)\\
&=1*P\left(a_1,\ldots,a_n|I^T\right)\\
&=\sum_{o_i,m_i}\left[P\left(a_1,\ldots,a_n|o_1,\ldots o_n,m_1,\ldots,m_n\right)*P\left(o_1,\ldots o_n,m_1,\ldots,m_n|I^T\right)\right]\\
&=\sum_{o_i,m_i}\left[P\left(a_1,\ldots,a_n|o_1,\ldots o_n,m_1,\ldots,m_n\right)*\prod_{i=0}^{n}P\left(o_i|I^T\right)*\prod_{i=0}^{n}P\left(m_i|I^T\right)\right]
\end{aligned}
$$

We estimate the probability *P(a1, a2, ..., an | o1, o2, ..., on, m1, m2, ..., mn)* using the linear regression with binary features *[o1, o2, ..., on, m1, m2, ..., mn]*. We assume the *P(a1, a2, ..., an | o1, o2, ..., on, m1, m2, ..., mn)* to be a linear function of *[o1, o2, ..., on, m1, m2, ..., mn]*. That is,

$$
\begin{aligned}
E\left[P\left(a_1,\ldots,a_n|o_1,\ldots o_n,m_1,\ldots,m_n\right)\right]&\\
&=E\left[a_1,\ldots,a_n|o_1,\ldots o_n,m_1,\ldots,m_n\right]\\
&\approx\alpha+\beta\left[o_1,\ldots o_n,m_1,\ldots,m_n\right]+\gamma\left[o_1,\ldots o_n,m_1,\ldots,m_n\right]^2
\end{aligned}
$$

where we estimate parameters α, β, and γ through training. Number of training samples need for training *P(a1, a2, ..., an | o1, o2, ..., on, m1, m2, ..., mn)* is *O(n)*.

4.3. Error handling

In this sub-section, we discuss how our algorithm handles errors from the segmentation layer, the object layer, and the motion layer. The semantic layer must have power to recover from the failures of the segmentation layer, the object layer, and the motion layer. The segmentation layer may fail to track to objects, the object layer may misclassify objects, and the motion layer may misestimate object motions. The power to handle errors of the lower layers is not only important for the reliably recognition but also for providing feedback. The semantic layer must provide feedback to the layer that made a failure, telling the layer to "rethink" about the decision it made. In this sub-section, we present how our recognition algorithm handles those failures and recognize human-object interactions reliably.

Tracking failure: The segmentation layer may fail to track objects because of occlusions. For example, when a person is stealing another's suitcase, the segmentation layer may label the suitcase that is being stolen by the thief as a different suitcase than the initial one. The tracking failure can be crucial, since our time interval detection algorithm in 4.1 assumes that all objects are tracked correctly.

In order to cope with tracking failures, we modified the algorithm presented in sub-section 4.1 and 4.2. Previously, the algorithm checks whether the object in one frame is identical to the object in another frame solely based on tracking results. Now, when detecting the time intervals, the algorithm considers the object match between the sub-events probabilistically. That is, in the case of the stealing interaction, the system calculates the probability that the suitcase that owner was carrying is identical to the suitcase that the thief is carrying later. If we denote objects in sub-events that have to be identical as (oi^1, oi^2, …, oi^k), then the probability of the activity is as follows.

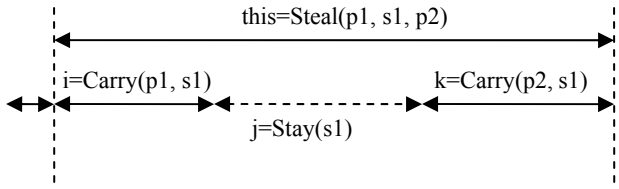


Figure 4: Example of the 'hallucination' for recognizing human-object interaction 'steal (p1, s1, p2)'. The sub-event 'stay' of the suitcase was not detected because of error. The dotted arrow indicates the hallucination to help system recognize person2 stealing person1's suitcase.

$$
\begin{aligned}
&P\left(R^{\langle s,e\rangle}|I^T\right)\\
&=\sum_{o_i,m_i}\left[\begin{array}{l}P\left(a_1,\ldots,a_n|o_1,\ldots o_n,m_1,\ldots,m_n\right)*\\ \prod_{i=0}^{n}P\left(o_i|I^T\right)*\prod_{i=0}^{n}P\left(m_i|I^T\right)*\prod_{i=0}^{n}P\left(o_i^1=o_i^2=\ldots o_i^k\right)\end{array}\right]
\end{aligned}
$$

Object recognition failure: The algorithm presented in sub-section 4.1 and 4.2 is able to handle object recognition failures without any modification. Basically, any object with non-zero probability of being classified into the desired category will be considered a candidate participant of the activity. If a time interval is detected treating a misclassified object as a participant, the probability of the activity containing the misclassified object will be calculated accordingly using mechanisms presented in 4.2. If overall probability is high enough, the system recognizes the activity even with the failure of the object layer.

Motion recognition failure: Section 4.1 presented how the system calculates the candidate time intervals which satisfy the language-like representation constructed for the activity. In section 4.2, a mechanism to associate probability with each time interval is shown. However, the process presented in both sections relies on the fact that the motion layer detects time intervals correctly, at least with very low probability. That is, the system presented in 4.1 and 4.2 shows how to recognize a high-level activity even with some of sub-events having a low probability, but it does not show how to overcome the complete failure of the motion layer. If the motion recognition fails completely, so even a local maximum having very small probability does not exist, then our time interval detection algorithm cannot detect any time intervals for a high-level activity which has that motion somewhere below the hierarchy tree.

We introduce the concept of 'hallucination' similar to [2] to overcome the complete failure of the motion layer. The hallucinations are time intervals of object motion that are

inserted by the system, even when no motion was detected in the time intervals. The role of hallucinations is to complement the failure of the motion layer, by making the semantic layer think as if there exists a correctly detected motion of an object. We normally insert hallucinations between nearby sub-events. The probability associated with hallucinations is set to a very low value. Figure 4 shows an example of the hallucination.

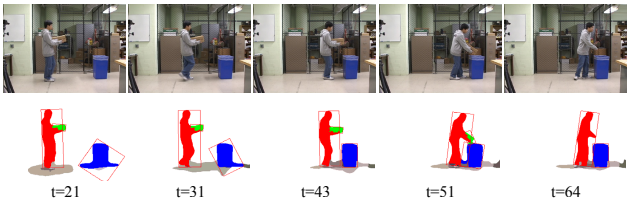
5. Experiments

We tested our framework to recognize meaningful activities in an airport-like environment. Four categories of objects participated in the activity: humans, suitcases, boxes, and trash bins. Object motions are placed into five classes: move left, move right, move upward, move downward, and stay stationary. Based on these four categories of objects and five classes of motions, high-level human-object interactions are represented and recognized. 'a person carrying a suitcase', 'a person leaving his/her suitcase', 'a person stealing another's baggage', 'a person carrying a box', 'a person leaving a box', and 'a person placing a box into the trash bin' are six high-level human-object interactions that our system recognizes.

A Sony VX-2000 camcorder is used to record videos of human-object interactions. The video was taken at 320*240 pixel resolution with 15 frames per second. The system is implemented in C++ in the Windows platform. There were 45 sequences of images total (over 2000 frames), each sequence containing more than one human-object interaction. As a result, dataset contains total 80 high-level interactions, and each human-object interaction was taken at least 10 times. The object layer and the motion layer are trained with 5 sequences randomly drawn from the total set. In the semantic layer, the high-level representation of human activities is constructed by human expert. The parameters in the regression part are initialized with domain knowledge, and updated with chosen 5 sequences. The system was tested for entire dataset, recognizing objects, motions, and activities.

Figures 5 and 6 presents the segmentation result, the object recognition result, the motion recognition result, and the final recognition result of human-object interactions 'a person placing a box into the trash bin' and 'a person stealing another's baggage'. The segmentation between the objects (humans, boxes, trash bins, and suitcases) was done correctly. Objects were classified and their motions were estimated accordingly. Time intervals associated with the motion estimation results are also illustrated. Finally, the figure presents time intervals associated with the recognition results of high-level human-object interactions 'a person placing a box into the trash bin' and 'a person stealing another's baggage.

The recognition accuracy of six human-object interactions is presented in Table 1. In order to illustrate the



Object recognition results:
(numbered from left to right)
Object #1: Human 0.99 Trash bin 0.01
Object #2: Box 0.98 Suitcase 0.02
Object #3: Trash bin 0.93 Suitcase 0.07

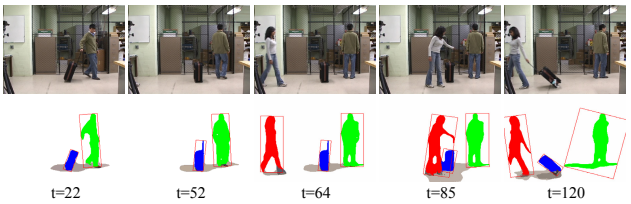
Motion recognition results

MoveLeft(o1)	[4, 19], [23,40]
Stay(o1)	[1, 4], [19, 23], [40, 64]
MoveLeft(o2)	[10, 16], [21, 32], [37, 41]
Stay(o2)	[17, 21], [33, 37], [41, 47]
MoveDown(o2)	[47, 52]
Stay(o3)	[1, 64]

Final human-object recognition results

Carry(o1, o2)	[10, 16], [21, 32], [37, 40]
Trash(o1, o2, o3)	[37, 52]

Figure 5: Example recognition results of the human-object interaction 'person placing a box into the trash bin'. The object recognition, the motion recognition, and all intermediate human-object interaction recognition results are presented.



Object recognition results:
Object #1: Human 0.99 Trash bin 0.01
Object #2: Suitcase 0.83 Trash bin 0.17
Object #3: Human 0.98 Trash bin 0.02

Final human-object recognition results

Carry(o3, o2)	[22, 25]
Stay(o2)	[26, 76], [82, 90], [96, 104]
Carry(o1, o2)	[107, 117]
Steal(o3, o2, o1)	[107, 117]

Figure 6: Example recognition results of the human-object interaction 'a person stealing another's baggage'.

power of our new probabilistic recognition algorithm, the recognition result of the system with probabilistic error-handling algorithm is compared to the system without it. The number of atomic actions and spatial relationships that compose the interactions is also listed to show the complexity of the activity. The rate of 'true positives' are shown in Table 1, while false positive rates were omitted because they were almost 0. The result shows that our new algorithm enables the system to recover the failures of the object recognition or motion estimation, especially in case of complicated activities. Particularly, the recognition rate of the non-probabilistic algorithm for the activity 'a person placing a box into the trash bin' was low because the

motion layer was not able to reliably estimate one of its sub-events, 'move down' motion of a box (motion estimation accuracy 0.6). Our probabilistic semantic layer compensated for such failure with a hallucination generated based on the other sub-events, acquiring significantly higher recognition rate. As the number of sub-events increases, the chance of compensating increases which results the high-level activities to be recognized reliably.

In addition, we also conducted experiments to show that the feedback from the accurate semantic layer improves the performance of the object recognition and the motion estimation. The experimental results justify our approach to integrate the result of object recognition, the result of motion recognition, and the feedback from the semantic layer to help the object recognition and motion estimation.

Table 2 shows the performance of our object recognition. Final recognition performance of the object layer is compared with that of the primitive object module, which recognizes objects solely based on the input features. We were able to observe that our primitive module tends to confuse suitcases and trash bins because of the similarity between their shapes. The final classification decision made by the object layer overcame this problem by taking advantage of the recognition results of the other layers. The table clearly illustrates that accuracy of the object layer increases as a result of the compensation.

The recognition accuracy of objects' motion is illustrated in table 3. Because of the shadow changes, the primitive motion module was weak on estimating 'staying' motion. Also, the primitive motion module sometimes fails to estimate 'move down' motion of a box due to occlusions between a human and the box. Table 3 shows that our system was able to compensate those failures with the help of the object layer and the semantic layer.

6. Conclusion

We presented a novel framework for the reliable recognition of high-level human-object interactions. The framework integrates the object recognition, the motion estimation, and the semantic-level recognition of high-level human-object interactions. Each layer probabilistically compensates for the failure of the layer with the use of the decisions made by the other layers. The experiments show that our framework not only results in the reliable recognition of high-level human-object interactions, but also increases the accuracy of object recognition and motion estimation.

The main technical contribution made in this paper is on the probabilistic semantic layer to hierarchically recognize high-level human-object interactions. An algorithm to reliably recognize human activities represented in terms of complicated temporal, spatial, and logical relationship has not been developed before. Our algorithm probabilistically recognizes complicated human-objects interactions even when the object recognition or the motion estimation component made failures. Error handling mechanisms for failures of the other components were analyzed in detail.

Interaction	**# of atomic action**	**# of spatial relation**	**Algo. w/o prob.**	**Algo. with prob.**
Carry(p1,s1)	2	1	0.866	0.933
Leave(p1,s1)	3	1	0.8	0.9
Steal(p1,s1,p2)	5	2	0.7	0.9
Carry(p1,b1)	2	1	0.9	0.95
Leave(p1,b1)	3	1	0.7	0.8
Trash(p1,b1,t1)	3	3	0.4	0.9
total			**0.778**	**0.911**

Table 1: Overall recognition accuracy of the system

Object	**primitive**	**final**
Human	0.937	0.957
Suitcase	0.895	0.946
Box	0.952	0.971
Trash bin	0.883	0.982
total	**0.918**	**0.957**

Table 2: Object recognition accuracy

Motion	**primitive**	**final**
Move left (right)	0.957	0.985
Move down (up)	0.6	0.85
Stay	0.794	0.941
total	**0.856**	**0.952**

Table 3: Motion estimation accuracy

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] D. Minnen, I. Essa, T. Starner, "Expectation Grammars: Leveraging High-Level Expectations for Activity Recognition," *CVPR'03*, p. 626, 2003.

[3] D. J. Moore, I. A. Essa, and M. H. Hayes III. Exploiting human actions and object context for recognition tasks. ICCV, volume 1, pages 80-86, Corfu, Greece, 1999

[4] I. Haritaoglu, D. Harwood, and L. S. David, 2000. W4: Real-Time Surveillance of People and Their Activities. *IEEE Trans. on PAMI.* 22, 8 (Aug. 2000), 809-830

[5] 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.

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

[7] 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

[8] M. S. Ryoo and J. K. Aggarwal, "Recognition of Composite Human Activities through Context-Free Grammar Based Representation," *CVPR'06*, pp. 1709-1718, 2006.

[9] T. Starner, A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," *ISCV*, p. 265, 1995.