Vol.2009-CVIM-167 No.3
2009/6/9

Modeling and Recognizing Human Activities from Video

Kris M. Kitani[†1,†2] and Yoichi Sato[†2]

This paper presents a complete computational framework for discovering human actions and modeling human activities from video, to enable intelligent computer systems to effectively recognize human activities. A bottom-up computational framework for learning and modeling human activities is presented in three parts. First, a method for learning primitive actions units is presented. It is shown that by utilizing local motion features and visual context (the appearance of the actor, interactive objects and related background features), the proposed method can effectively discover action categories from a video database without supervision. Second, an algorithm for recovering the basic structure of human activities from a noisy video sequence of actions is presented. The basic structure of an activity is represented by a stochastic context-free grammar, which is obtained by finding the best set of relevant action units in a way that minimizes the description length of a video database of human activities. Experiments with synthetic data examine the validity of the algorithm, while experiments with real data reveals the robustness of the algorithm to action sequences corrupted with action noise. Third, a computational methodology for recognizing human activities from a video sequence of actions is presented. The method uses a Bayesian network, encoded by a stochastic context-free grammar, to parse an input video sequence and compute the posterior probability over all activities. It is shown how the use of deleted interpolation with the posterior probability of activities can be used to recognize overlapping activities. While the theoretical justification and experimental validation of each algorithm is given independently, this work taken as a whole lays the necessary groundwork for designing intelligent systems to automatically learn, model and recognize human activities from a video sequence of actions.

1. Introduction

While the computing power of computers and the number of sophisticated algorithms has grown, current research is far from closing the so called *semantic gap* between physical observations and semantic expressions. Despite the seemingly insurmountable task of teaching a computer to learn as we do, the aim of this work is to propose a computational framework for automatically discovering, modeling and recognizing human activities.

In this paper the process of understanding human activities is divided into three sub-tasks: (1) learning primitive actions (Section 2), (2) discovering the structure of activities (Section 3) and (3) recognizing structured activities (Section 4). As such, this paper presents a bottom-up computational framework activity analysis by presenting the entire system in consecutive stages, where the output of each task provides the input of the proceeding task.

The three contributions of this work are as follows:

- a bimodal learning approach that uses both **motion and visual context without the use of *a priori* scene knowledge** to learn primitive interactive actions, whereas previous work used only motion or relied on *a priori* knowledge of the appearance of objects or actors.
- a **new unsupervised algorithm for learning syntactic structure from noisy data** (potentially all negative examples), whereas previous work on grammatical induction used only training sets of positive examples.
- a probabilistic syntactic framework for **robustly recognizing overlapped human activities**.

2. Learning action primitives

In this section, an unsupervised method for learning primitive actions from a corpus of actions is proposed. It is shown that action categories can be discovered effectively when both motion and visual appearance are used to represent primitive action. Primitive actions are defined here as humans actions that can be recognized over a very short period of time (a few seconds). For example, grabbing a cup, typing on a keyboard or flipping the page of a book can be recognized within a few seconds of observing the action. Learning primitive actions are important because they are the basic building blocks of many high-level activities[12),17),23)].

Supervised learning techniques using such models as HMMs[14),38)], Bayesian classifiers[36)] and temporal dynamics[37)] have been successful in describing primitive actions but require labeled data or a considerable amount of prior knowledge.

†1 University of Electro-Communications
†2 University of Tokyo

Vol.2009-CVIM-167 No.3
2009/6/9

Fig. 1 Leveraging visual features for action recognition: Relevant visual features (green) induced by using the telephone and irrelevant features (purple) produced by unrelated background objects.

Recently, an approach of growing interest for unsupervised action discovery is the use of *generative latent variable models*[2),13),29),40)] based on the bag-of-words paradigm, originally developed for topic discovery from text.

Niebles[28)] proposed the application of a generative model to video to learn action categories (topics) from a bag-of-features. They used exactly the same framework as 13) by simply replacing document indices with video indices, and words with spatial-temporal (ST) volumes. Their approach showed that similar to text, the local features of an action can be treated as though they were *exchangeable* (an action can be treated as a bag of uncorrelated features) to learn action categories. However, the conceptual problem with a straightforward use of a language model for action discovery is that the models are uni-modal (e.g. use only words).

It is known from experience that actions are composed of motions and visual appearance. For example, the hands of a person playing a piano and typing on a keyboard might have very similar motions but can easily be differentiated using the visual context of a piano or a keyboard. In fact, findings from neural science make it clear that actions are mentally perceived as a mix of motions and visual features of present objects[9)]. In the light of this fact, many previous approaches to action discovery are limited by the fact that they only consider one mode, namely, motion[28),43)].

While the joint use of appearance and motion to describe action is not entirely new, this algorithm differs from previous algorithms in that the proposed method does not use *a priori* information about the category, shape, size or color of actors or objects in the scene[10),11),24),27)]. Presented in this section is a robust framework for primitive action discovery by leveraging both motion and relevant visual context without the use of *a priori* information (e.g., an explicit shape model or pre-defined object categories). Experiments show that the proposed method properly leverages relevant visual appearance and is robust against irrelevant visual features (Figure 1) when learning action categories.

2.1 Proposed method for learning action primitives

The goal is to learn the primitive action categories that occur within a video corpus. First temporal features and spatial features are extracted from each video segment, under the assumption that actions are defined by both temporal motion and visual context. Then a description of a dimension reduction scheme is given to create a codebook for each feature type . Finally, an explanation of a bi-modal generative model is presented, that uses the histograms produced from a video corpus to learn the latent action categories.

For each frame in the training corpus, a sparse set of spatial features is extracted by finding SIFT key points[20)]. Likewise, a sparse set of temporal features are extracted from the video frames by extracting a $7 \times 7 \times 4$ (a 7×7 spatial window over 4 frames) spatiotemporal volume[3)] for pixels that detected as a good feature to track[35)] and are tracked by optical flow[4)] for two consecutive frames. More complex temporal keypoints can also be used, such as spatiotemporal cuboids[8)] or space-time interest points[18)].

Compared to documents or images, the number of features that can be extracted from a video sequence can be very large (e.g., about 20 million temporal features for 7 minutes of video). Therefore, an efficient two stage clustering process that combines an online and offline algorithm is implemented to process the descriptors generated by the video corpus.

An online clustering algorithm termed *nearest representative point clustering* (NRPC) is used to cluster descriptors and generate a histogram for all the videos in one pass. The NRPC algorithm is given as follows.

for every video segment d in the corpus **d do**
  Initialize segment histogram $\mathbf{v}_d = \mathbf{0}$
  for every descriptor $\mathbf{x}_{di}$ extracted from segment d **do**
    Find nearest representative point $\mathbf{c}_j$ to $\mathbf{x}_{di}$
    if $L_2(\mathbf{x}_{di}, \mathbf{c}_j) > \theta$ **then**
      Create new representative point $\mathbf{c}_k \leftarrow \mathbf{x}_{di}$
      Initialize count of centroid $v_{dk} = 1$
    else
      Increment count v_{dj} of nearest representative point $\mathbf{c}_j$
    end if
  end for
end for

The NRPC algorithm takes a single descriptor $\mathbf{x}_{di}$ from the set of all descriptors extracted from segment d and decides whether to update the count of a pre-existing cluster or create a new cluster, depending on a threshold θ. After processing all descriptors, a set of n clusters $\mathbf{c}_1, \ldots, \mathbf{c}_n$ and a corresponding n dimensional histogram vector of counts $\mathbf{v}_d = (v_{d1}, \cdots, v_{dn})^T$ for the video segment d are obtained.

Each video segment $d \in \mathbf{d}$ is processed in the same way to produce the set of $m = |\mathbf{d}|$ histogram vectors of the histogram matrix $\mathbf{V} = (\mathbf{v}_1, \cdots, \mathbf{v}_m)$. Notice that the number of clusters n can potentially increase each time a new video is processed (i.e., new clusters are created). The histograms of previously processed videos are simply padded with zeros to keep the same dimensionality n. This clustering process is done once for each feature modality (i.e., spatial features and temporal features). This type of online clustering is effective for video because many nearly identical features are produced by a single action.

In the second stage, the dimensionality of the training data is further reduced using a more holistic approach called Non-negative matrix factorization (NMF)[19]. NMF decomposes the $n \times m$ histogram matrix $\mathbf{V}$ (each column is a histogram of descriptors for a video) into a $n \times r$ basis matrix $\mathbf{W}$ and the $r \times m$ encoding matrix $\mathbf{H}$, such that $\mathbf{V} \approx \mathbf{WH}$. NMF is executed twice independently, once for spatial features and once for temporal features.

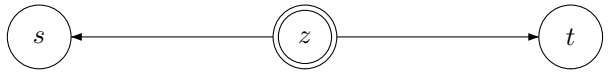

Fig. 2 Bi-modal latent variable model defined by the latent topic z, a spatial feature s and a temporal feature t.

2.2 Merging motion and visual context via the action model

The proposed model (Fig. 2) is a bi-modal expansion of the standard mixture of unigrams model[29] that defines the probability of a video segment $d \in \mathbf{d}$ as below,

$$p(d) = \sum_z p(d|z)p(z) \tag{1}$$

$$p(d|z) \propto \prod_s p(s|z)^{n(s,d)} \prod_t p(t|z)^{n(t,d)} \tag{2}$$

where the term $n(s,d)$ represents the number of times a spatial feature s has occurred in a video segment d. The term $n(t,d)$ is interpreted similarly for temporal features. The parameters $p(s|z)$, $p(t|z)$ and $p(z)$ are learned by maximizing the log-likelihood of the entire video corpus $\mathbf{d}$,

$$\log p(\mathbf{d}) = \sum_d \log \sum_z \Big[\prod_s p(s|z)^{n(s,d)} \prod_t p(t|z)^{n(t,d)} \Big] p(z) \tag{3}$$

. In the expectation step, the posterior of the latent variable is computed using Bayes' rule.

$$p(z|d) = \frac{p(d|z)p(z)}{\sum_{z'} p(d|z')p(z')} \tag{4}$$

In the maximization step, the updates are computed.

$$\hat{p}(s|z) \propto \sum_d n(s,d)p(z|d) \tag{5}$$

$$\hat{p}(t|z) \propto \sum_d n(t,d)p(z|d) \tag{6}$$

$$\hat{p}(z) \propto \sum_d p(z|d) \tag{7}$$

This process between the expectation step and the maximization step is repeated until the log-likelihood function converges at a local optima.

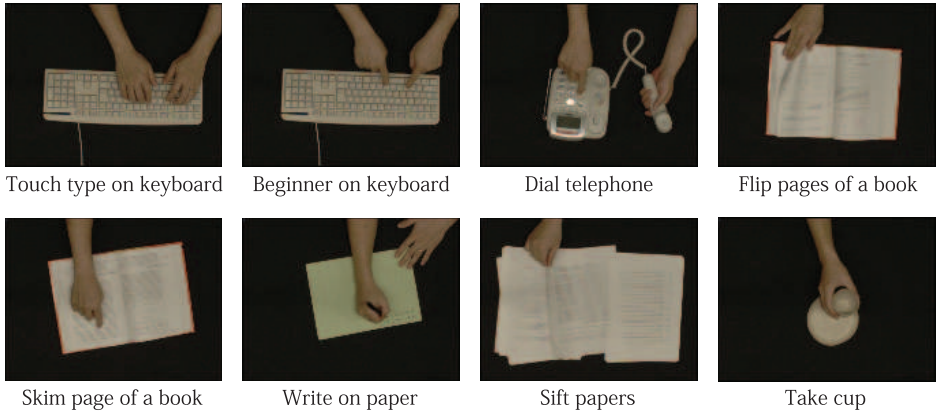

Fig. 3 The first motion and object corpus C_{OBJ} consists of 8 different primitive actions that involve a related physical object.

2.3 Action datasets

Publicly available datasets used for human action recognition, like the KTH dataset[34], have very little background variation (i.e., a wall or a field) and usually only involves an actor with no interactive objects[8),44)]. In contrast, it is reasonable to assume that many other objects will be visible in real world videos of human actions, especially important visual features that help define the actions being performed. Three primitive action (interaction) video datasets are presented in Figures 3, 4 and 5 and are used to show how the proposed method is able to leverage relevant visual context along with motion information to effectively discover action categories. For each datatset, there are five video segments per action and each action video segment is a three-second interval randomly spliced from the original video. All videos were created at a resolution of 160×120.

2.4 Experiments with hand action datasets

First a baseline experiment is performed using only temporal features as in 28). Then three experiments are performed using the proposed framework and it is shown how leveraging visual context improves learning performance. The standard AUC measure is given along with the probability of correct categorization (PCC) which represents the degree to which a dataset is properly categorized. All the results are summarized in Table 1 and more details regarding the implementation and evaluation measures can be found in 15).

Fig. 4 The second motion and background corpus C_{BG} consists of 9 actions composed from 3 different motions and 3 different background objects. Direction of motion is shown in white.

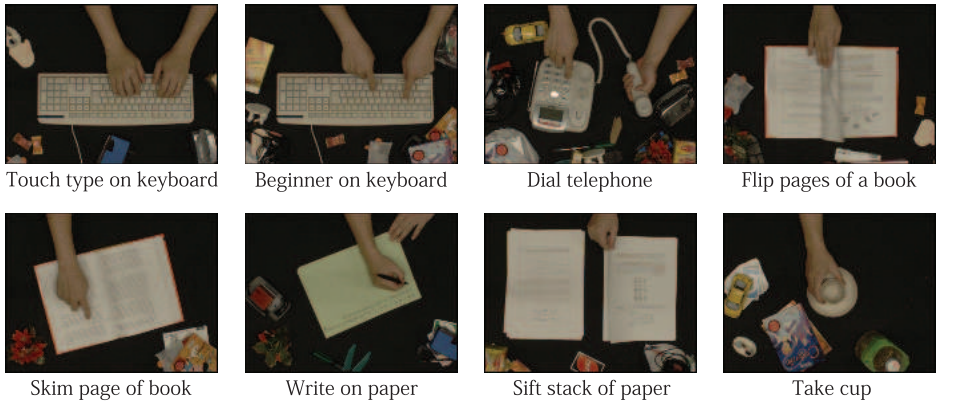

Fig. 5 The third motion with objects and background (messy desktop) corpus C_{BGOB} contains the same actions as the first corpus C_{OBJ} but also includes varied random background objects for each video segment.

The proposed method using both motion and visual context is utilized to learn action categories from the same motion and object corpus C_{OBJ}. In comparison to the strict use of only temporal features (Figure 6), it is observed from the bar graph (Figure 7) of the posterior probability that all actions contained in the video corpus have been accurately discovered with high confidence. Leveraging the visual appearance of the action and related objects significantly increased the confidence of classification performance.

Next the action and background corpus C_{BG} is utilized to test whether the

Table 1 PCC and AUC values for differing configurations.

	Temporal		Spatial		Bimodal	
Dataset	PCC	AUC	PCC	AUC	PCC	AUC
C_{OBJ}	85.12%	1.000	92.18%	1.000	99.05%	1.000
C_{BG}	44.96%	0.878	91.35%	1.000	92.29%	1.000
C_{BGOB}	77.32%	0.998	82.68%	1.000	93.62%	1.000

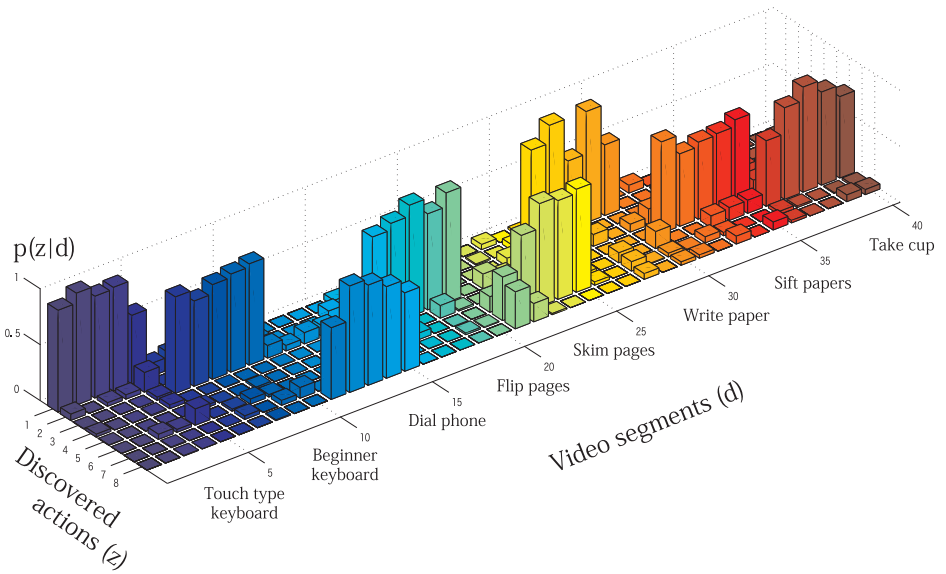

Fig. 6 Baseline results using only **temporal** features for corpus C_{OBJ}. The horizontal axes gives the ground truth for each video d and the discovered action category z. The vertical axis is the posterior probability $p(z|d)$.

proposed method is able to distinguish between actions with very similar (same) motions, that can only be differentiated by their visual context. Notice that each combination of visual context and motion have been correctly categorized with high PCC 92% (Table 1). Since there are only three differentiable motions in the database, the decomposition of the temporal feature histograms is very difficult resulting the low PCC of 44%.

In reality, primitive actions occur in various types of visual contexts and it is important to be able to leverage only the relevant visual features that should

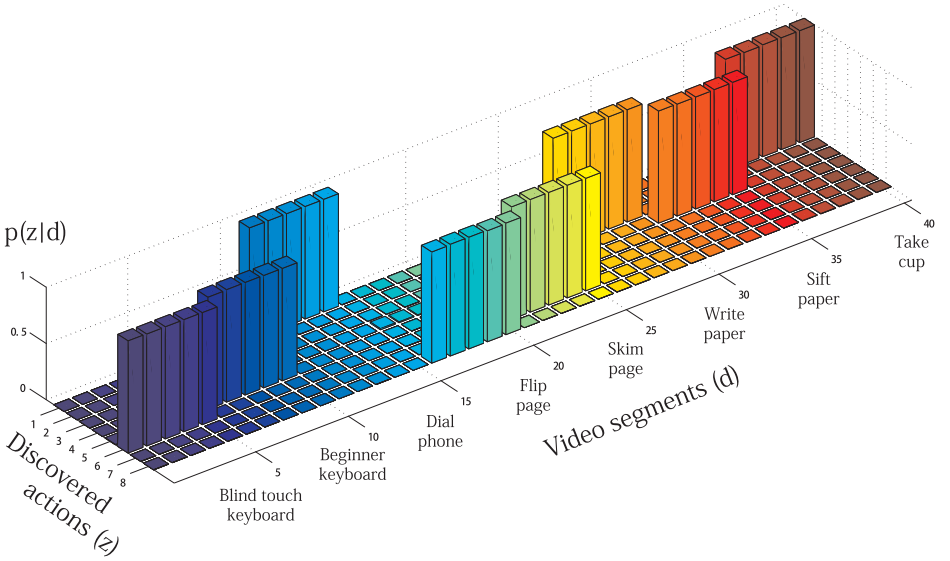

Fig. 7 Posterior probabilities using the proposed **bi-modal** method with the corpus C_{OBJ}, which contained 8 different actions.

be associated with an action (Figure 1). In the last experiment, the proposed method is applied to the motion with object and background corpus C_{BGOB} and it is shown how the proposed method can leverage relevant visual features to discover actions categories, even with various cluttered backgrounds (visual noise). Results show that the proposed approach is able to learn the actions of the corpus C_{BGOB} with an average PCC of 93.62%.

2.5 Summary

A novel framework for discovering action categories by leveraging relevant visual context and motion features has been presented in this section. In the proposed framework, a fast two stage clustering algorithm was implemented via nearest representative point clustering and non-negative matrix factorization, to generate a term-by-document matrix as the input to the bimodal mixture model. The bi-modal mixture model used both visual features and temporal features to discover latent action categories. Through the experiments it was shown that the proposed approach is able to accurately classify actions by leveraging rele-

vant visual appearance to disambiguate similar motions. It was also shown that the proposed method is robust against irrelevant visual features generated by the background while at the same time leveraging relevant visual features to accurately discover primitive action categories.

3. Learning the structure of activities

In this section a new framework for discovering the basic temporal structure (grammar) within an action symbol string is proposed. To find the optimal grammar in an information-theoretic sense, the minimum description length principle is used to identify a set of primitive actions that defines an optimal stochastic context-free grammar (SCFG). The SCFG is a model that has been widely utilized for natural language processing and in recent years, has also been shown to be effective in modeling human activities extracted from video[14),22),23),33)]. Other non-hierarchical sequential state-based models (finite-state automata, hidden Markov models, n-grams, etc.) have also been successfully applied to human activity recognition but are limited by the fact that they do not explicitly describe the hierarchical structure of human activities.

One important task involved in using an SCFG for activity analysis is the task of *learning* the grammar. In fact, all of the aforementioned work uses manually designed grammars created from the *a priori* knowledge of the target domain and thereby avoid the issue of grammar learning. In comparison to work that used pre-defined grammars, research dealing with the issue of automated learning has been minimal and always assumes a pure data set for learning. Wang[42)] used an experimental scenario similar to Ivanov and implemented HMMs to produce primitive action symbols from a video segment of a conductors hand motions. The primitive actions produced by the HMMs were then fed into a pre-existing CFG learning algorithm COMPRESSIVE[25)] to learn the activity grammar. Due to the fact that COMPRESSIVE requires positive examples to generate the CFG, it can be shown that their system is very sensitive to noise in the input symbol string. While a noise-less input stream may be a reasonable assumption when learning a grammar from a string of words, it is a naive assumption when attempting to learn an activity grammar from a symbol string produced by stochastic detectors from a highly variable action sequence created by human actors.

In contrast to previous works, this section proposes a new grammar learning method that works even in the presence of noise. The proposed method places an assumption of noise on different combinations of terminal symbols and tests that assumption using the minimum description length (MDL) principle. Then using the results of the MDL evaluation, the proposed method finds the best set of terminal symbols that yields the most compact and descriptive activity grammar.

3.1 Preliminaries

When considering the task of learning an activity from a string of action symbols, it is reasonable to expect different types of noise that might hide the basic structure of the activity that is to be learned. The first type of noise is inherent to human activities which is termed here as *inherent noise*. Inherent noise is caused by superfluous actions that do not play an important role in defining the activity to be learned. These secondary action symbols (noise symbols) tend to appear with irregular frequency and order, and fill in the gaps between the important action symbols. The second type of noise is *system noise* caused by the instability of the image processing system. System noise can be attributed to changes in appearance that cause the image processing system to insert, substitute or delete (miss) random symbols from the symbol string. Symbols that are inserted, substituted or deleted with a high frequency should not be used for learning because they introduce much randomness to the symbol string. While the primary assumption is that of inherent insertion noise, it is also shown in section 3.4 how the proposed method also shows robust performance against system noise when these assumptions are violated.

As mentioned before a context-free grammar (CFG) is used here to model human activity because of its ability to explicitly and compactly describe hierarchal structure. A CFG is defined by the 4-tuple $\mathbf{G} = \{\mathbf{T}, \mathbf{N}, S, \mathbf{R}\}$, where $\mathbf{T}$ is a finite set of terminal symbols, $\mathbf{N}$ is a finite set of non-terminal symbols, S is the start symbol (a special non-terminal symbol) and $\mathbf{R}$ is the set of production rules. The production rules take the form $A \rightarrow \lambda^*$, which states that non-terminal symbol A produces the string λ^* of one or more symbols. When a probability $p(A \rightarrow \lambda^*)$ that satisfies the condition $\sum_i p(A \rightarrow \lambda_i^*) = 1$, is associated to each rule, the grammar becomes a stochastic context-free grammar (SCFG).

3.2 Proposed method

3.2.1 Setting up noise hypothesis

To learn the true grammar from noisy training data, the noise must be removed from the training data. However, since it is not know *a priori* which symbols are noise, it is proposed to set up various hypothesis (noise or not noise) against each unique primitive symbol and evaluate the assumptions using a MDL criterion. Formally, given the training data $\mathbf{W} = \{W_1, \ldots, W_l\}$, a concatenation of l activity sequences W_i, where each activity sequence $W_i = \{w_1, \ldots, w_p\}$ is a string of primitive action symbols $w_j \in \mathbf{T}$, the objective is to identify the symbols that are not useful (noise) for learning the true grammar.

A single hypothesis divides the set of primitive actions (terminal symbols) into two sets: the set of noise symbols $\mathbf{w}^f = \{w_1^f, \ldots, w_v^f\}$ and the set of non-noise symbols $\mathbf{w}^t = \{w_1^t, \ldots, w_u^t\}$. Next, an initial grammar is constructed to reflect the hypothesis.

The first rule of the form $S \rightarrow \mathbf{W}'$ is the start production rule. S is a nonterminal symbol that represents all possible symbol strings produced by the grammar and in the initial stage $\mathbf{W}'$ is the concatenated training data encoded by the other production rules of the initial grammar. To attain the encoded input symbol string $\mathbf{W}'$, a plain input symbol string $\mathbf{W}$ is encoded to reflect the presuppositions made about each terminal symbol. This is done by replacing each terminal symbol w_i with the appropriate nonterminal symbol using the preterminal production rules, which are defined next.

The set of production rules of the form $N_i \rightarrow w_i^t$ is created for each presupposed non-noise symbol, where w_i^t is a non-noise terminal symbol and N_i is a newly created nonterminal. These preterminal rules effectively preserve the unique identity of the symbol in the training data.

The set of generic preterminal production rules of the form $\eta \rightarrow w_j^f$ is created for each noise terminal symbol, where w_j^f is a noise terminal symbol and the nonterminal η is a generic nonterminal representing all noise symbols. The generic absorption rule $\eta \rightarrow \eta\ \eta$ is also created, which encodes a series of adjacent noise symbols.

3.2.2 Learning the hypothesis grammar

Now that the presuppositions on the primitive action symbols have been encoded into the initial grammar, the next step is to learn the *hypothesis* grammar. The heuristic CFG learning algorithm COMPRESSIVE[25] is implemented to learn the hypothesis grammar. Upon completion of grammatical induction the string $\mathbf{W}''$ is reverted back to its original l activity sequences and sequences that have the same structure are grouped together and left the grammar $h \leq l$.

3.2.3 Testing using the MDL principle

The next goal is to find a hypothesis on the primitive action symbols that yields both a *compact* yet *expressive* grammar that describes the input symbol string. Reworded in the framework of MDL, the goals it is find an optimal selection of non-noise symbols that will yield a grammar $\mathbf{G}$ that minimizes the sum of the description length of the grammar $DL(\mathbf{G})$ and the description length of the data encoded by the grammar $DL(\mathbf{W}|\mathbf{G})$ (data log-likelihood).

$$\hat{\mathbf{G}} = \arg\min_G \{\ DL(\mathbf{G}) \ + \ DL(\mathbf{W}|\mathbf{G})\ \} \tag{8}$$

$$= \arg\min_G \{-\log P(\mathbf{G}) - \log P(\mathbf{W}|\mathbf{G})\}. \tag{9}$$

The encoding technique proposed by Stolcke[39] is implemented to find the description length of the grammar and the dynamic programming algorithm introduced by Pynadath[32] is implemented to calculate the description length of the data likelihood via inside probabilities.

The first term of the MDL equation is the description length of the grammar $DL(\mathbf{G})$. $DL(\mathbf{G})$ is a measure of the compactness of the grammar and is an indicator of the *regularity* found in the training data. Decomposing the prior over the grammar as the joint probability of the *parameters* θ_G and *structure* G_S gives

$$p(\mathbf{G}) = p(G_S, \theta_G) = p(\theta_G|G_S)p(G_S), \tag{10}$$

where the prior over the grammar parameters is a uniform Dirichlet distribution

$$p_N(\theta_G|G_S) = \frac{1}{B(\alpha_1, \ldots, \alpha_q)} \prod_{i=1}^{q} \theta_i^{\alpha_i - 1} \tag{11}$$

such that $\theta = (\theta_1, \ldots, \theta_q)$ is a multinomial distribution and B is a multinomial Beta function. Since there is no prior knowledge about the distribution of the

grammar parameters, the rule parameters θ_i and prior weights α_i are set to be uniform.

The structural probability $P(G_S)$ is calculated by

$$DL(G_S) = \sum_{R \in \mathbf{R}} \left(-\log p(r_R - 1; \mu) + r_R \log |\Sigma| \right). \quad (12)$$

where $r \log_2 |\Sigma|$ is the description length the r symbols in the grammar and $-\log p(r-1;\mu)$ is the log of a Poisson distribution which represents the description length of the length of a rule $R \in \mathbf{R}$. Further explanation and justification of the formulation of the description length of the grammar can be found in the original work[39].

It is not enough to evaluate the description length of the grammar because a grammar chosen purely based on grammar size will favor a very small grammar which may not explain the data well. The second term in the MDL equation is the description length of the data likelihood $DL(\mathbf{W}|\mathbf{G})$. $DL(\mathbf{W}|\mathbf{G})$ works to balance the effect of the first term by quantifying the expressive power of the grammar.

First, the data likelihood is calculated and then converted into a description length using Shannon's coding theory (negative log of the probability). The data likelihood is calculated using a chart of β probabilities created using the procedure outlined in the original work[32]. Once a chart has been constructed for a sequence $W = \{w_1, \ldots, w_{j_{max}}\}$, the data likelihood can be computed as a sum of β probabilities for all strings of length j_{max} produced by the root node S. Due to the insertion of abstraction rules when constructing the initial grammar and the possible creation of abstraction rules at post-processing, the maximum abstraction level k_{max} is two.

$$P(W_i|\mathbf{G}) = \sum_{k=1}^{k_{max}} \beta(S, j_{max}, k), \quad (13)$$

The total likelihood for all the sequences $\mathbf{W}$ is computed by equation (14) as a product of likelihoods for each sequence W_i. After the total likelihood has been computed, it is converted into a description length by taking the minus logarithm.

$$P(\mathbf{W}|\mathbf{G}) = \prod_{i=1}^{n} P(W_i|\mathbf{G}). \quad (14)$$

In summary, by calculating the description length of the grammar and the description length of the data likelihood, a framework for evaluating the quality of a presupposition made on the terminal symbols has been created. By identifying the hypothesis grammar that minimizes the total description length, the grammar that optimally describes the data is acquired.

3.3 Experiments with synthetic data

This section explores the conditions under which the proposed method is valid through experiments with synthetic data generated by a known grammar. Later it is also shown through an experiment with real data that the proposed method is able to produce intuitive results that aligns well with a human understanding of the target activity.

The synthetic data for each experiment was created using a pre-defined stochastic context-free grammar written according to a set of conditions. A set of d sample strings was generated by the artificial grammar and was used to analyze the proposed method. After the analysis, each hypothesis grammar was ranked according to its description length. Throughout this section, the grammar which uses the correct non-noise symbols is termed as the *true grammar* and use the rank of the grammar as a measure of the success of the proposed method. The desire is for the rank of the true grammar to always be first (i.e., the global solution of the MDL criterion).

3.3.1 Inherent insertion noise

Three different grammar parameters were varied to examine the performance of the proposed method to different types of inherent noise. First, three types of artificial grammars with different numbers of patterns were defined to evaluate the response of the proposed method to grammars with increasing complexity. Second, for each type of synthetic grammar, the number of terminal symbols were varied from 6 to 10. Third, to evaluate the effect of the sample size on the results, several training sets consisting of $d = 50, 150, 300, 500, 1000$ randomly produced strings were analyzed for each artificial grammar. The parameters and results for a subset of the artificial grammars are given in Table 2.

Table 2 Results with synthetic data (inherent insertion noise).

			$d = 50$	$d = 150$	$d = 300$	$d = 500$	$d = 1000$
Type	Non-noise	Noise	Rank of the true grammar				
1	3	3	3	1	1	-	-
1	3	4	3	1	1	-	-
1	4	5	11	4	1	1	1
1	4	6	14	4	1	1	1
1	5	4	34	15	5	1	1
1	5	5	54	15	5	1	1
2	3	3	11	4	1	1	-
2	3	4	12	4	1	1	-
2	4	5	28	11	5	1	1
2	4	6	65	13	5	1	1
2	5	4	91	43	16	6	1
2	5	5	242	34	16	6	1
3	3	3	23	5	1	1	-
3	3	4	28	5	1	1	-
3	4	5	102	43	11	4	1
3	4	6	213	89	10	3	1
3	5	4	87	80	30	16	5
3	5	5	181	136	27	17	5

The results show that the proposed method has identified the correct set of non-noise symbols when the sample size is sufficiently large (Table 2). Equivalently, the proposed method has been shown to produce sub-optimal results when the size of the training set was too small. In fact in the experiments with synthetic data, the true grammar was always outranked by smaller grammars when the sample size was insufficient. The results also show that complex grammars require more training samples than do simple grammars.

3.4 Synthetic system noise

Despite the fact that the method proposed thus far has been designed to address inherent insertion noise, it has been shown in preliminary experiments that the proposed method is also able to deal with system noise. In particular, the results show that the proposed method is able to cope with random insertion, deletion and substitution errors. Table 3 shows that the new modes of noise introduced by system noise increased the complexity of the task, which resulted in a need for more training samples to identify the true grammar.

3.5 Experiments with real data

A surveillance system in a local convenience store was setup to test the proposed method on real data. The system consisted of a single overhead CCD camera

Table 3 Results with synthetic data (inherent insertion and system noise).

			$d = 50$	$d = 150$	$d = 300$	$d = 500$	$d = 1000$
Type	Non-noise	Noise	Rank of the true grammar				
1	3	3	12	3	1	1	1
2	3	3	15	7	4	2	1
3	3	3	23	17	7	4	1

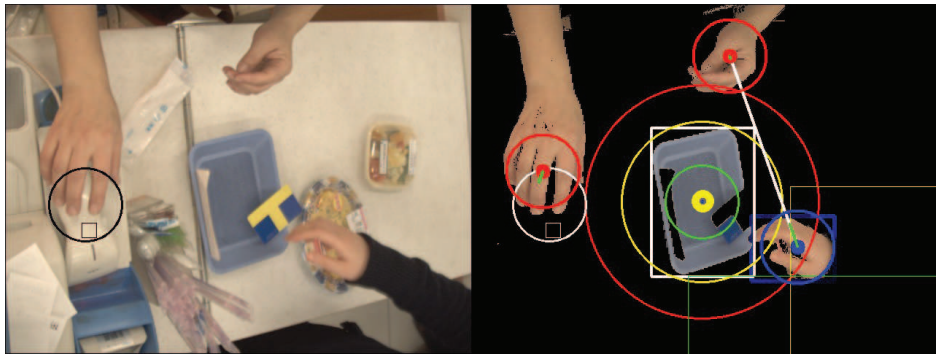

Fig. 8 Overhead view of the CCD camera mounted above the counter showing the results of the image processing to detect hands and tray.

(Figure 8) that captured the hand movements of the employee and the customer. In the experiment a total of more than 9700 frames were recorded and processed offline according to the proposed method. For this experiment primitive actions symbols were detected using simple image processing using application-specific domain knowledge for simplicity (see Figure 8). For this experiment a total of ten different types of primitive action symbols were extracted (see Table 4). A simple rule-based image processing system was implemented to extract the primitive action symbols in a top-down fashion. However, the proposed method will also work with any low-level image processing system that produces a string of primitive actions symbols.

A total of 369 symbols were automatically extracted from the convenience store surveillance video. The longest symbol sequence was eleven symbols long and the shortest sequence was three symbols long. Each sequence was concatenated into one long symbol string as the input to the proposed algorithm. The size of the

Table 4 Definition of the terminal symbols.

NO.	TERMINAL SYMBOL	DESCRIPTION
1	CUS_AddedMoney	Money found in tray after customer comes in contact with the tray
2	CUS_MovedTray	Customer moves tray
3	CUS_RemovedMoney	Customer removes money from tray
4	EMP_HandReturns	Employee hand returns after long absence
5	EMP_Interaction	Employee interacts with customer
6	EMP_MovedTray	Employee moves the tray
7	EMP_RemovedMoney	Employee moves money from tray
8	EMP_ReturnedScanner	Employee returns scanner
9	EMP_TookReceipt	Employee takes the receipt from the register
10	EMP_TookScanner	Employee picks up scanner

training data was $d = 55$ strings.

The grammar with the smallest overall description length was the hypothesis grammar that used the three symbols *EMP_ReturnedScanner*, *EMP_TookReceipt* and *EMP_TookScanner*. The grammar learned with these three symbols is given in Figure 9.

Notice that the symbols identified as non-noise symbols are all predictable actions performed by the employee. Since the employee has been trained to follow a certain protocol, his actions are predictable and ordered. In contrast, the actions of the customers show less regularity. Therefore, it makes sense that the MDL criterion identifies a grammar dependent only on the predicable actions of the employee as the optimal grammar.

3.6 Summary

This section has introduced a new method for acquiring the basic structure of an activity from a noisy symbol string produced by video. The proposed method placed presuppositions on each combination of terminal symbols and tested that hypothesis using an MDL criterion. The MDL equation measured the balance between a compactness and expressiveness of a grammar to encode the data, and provided a means of quantifying the quality of each presupposition. Experiments with both artificial and real data showed the proposed method is able to correctly identify an optimal grammar when the size of the training data was sufficient.

S → D				(0.02)	D → L	η		(1.000)
S → H				(0.16)	E → η	C		(1.000)
S → G				(0.18)	F → A	η		(1.000)
S → N	η			(0.04)	G → C	D		(1.000)
S → J				(0.13)	H → E	D		(1.000)
S → Q				(0.05)	I → η	B	η	(1.000)
S → η				(0.02)	J → C	F		(1.000)
S → N				(0.02)	K → η	D		(1.000)
S → R				(0.05)	L → F	B		(1.000)
S → J	B			(0.02)	M → C	η		(1.000)
S → M	L			(0.04)	N → E	A	B	(1.000)
S → M	A	H		(0.02)	O → E	η		(1.000)
S → C	K			(0.04)	P → E	I		(1.000)
S → C	A	M	F	(0.02)	Q → E	K		(1.000)
S → O	F			(0.02)	R → E	L		(1.000)
S → M				(0.02)				
S → O	L			(0.02)	η → η	η		(0.309)
S → O				(0.02)	η → CUS_AddMoney			(0.153)
S → P				(0.05)	η → CUS_MovedTray			(0.006)
S → I				(0.04)	η → CUS_RemMoney			(0.003)
S → K				(0.04)	η → EMP_HandReturn			(0.080)
A → EMP_ReturnedScanner				(1.00)	η → EMP_Interaction			(0.275)
B → EMP_TookReceipt				(1.00)	η → EMP_MovedTray			(0.028)
C → EMP_TookScanner				(1.00)	η → EMP_RemMoney			(0.147)

Fig. 9 Recovered optimal grammar using three non-noise symbols.

4. Recognizing structured human activities

In this section, a method for recognizing a string of primitive actions as an activity is introduced. The proposed method uses a weighted set of Bayesian networks, created from an underlying activity grammar, to detect activities occurring in the action symbol string[16].

According to findings in perceptual psychology[46], show that activities are perceived taxonomically and partonomically. At same time, activities can also be temporally overlapped or co-occur. For example, the transition of a person *walking through* a room might overlap with the activity of the person *departing* from the room. From the perspective of the system, it is difficult to identify the exact time at which the activity *walking through* has ceased and when the activity *departing* has started. Thus there is an inherent ambiguity at transitions between human activities which should be represented by a cognitive system.

The contribution of the proposed method described in this section lies in the novel application of deleted interpolation (DI) – a smoothing technique used in natural language processing – for recognizing temporally overlapped activities.

The majority of models that have been proposed for activity analysis are models that represent an activity as a sequential transition between a set of finite states (e.g. NDA[41], FSA[1], HMM[45], hybrid HMMs[30),31)]). However, due to the fact that most simple activities do not have complex hierarchical structure, these models have not explicitly incorporated the concept of hierarchy into the model topology.

There has also been other work that has proposed hierarchical models such as context-free grammars and hierarchical HMM to recognize structured activities[5),6),14),22),23),26)]. However, these models uses domains with high-level activities delineated by clear starting points and clear ending points, where the observed low-level action primitives are assumed to describe a series of temporally constrained activities (with the exception of Ivanov[14)]). However, in this section the focus is placed on a subset of human activities that have the possibility of being temporally overlapped. It is shown that these types of activities can be recognized effectively using the proposed framework.

4.1 Recognition system overview

The proposed recognition system consists of three major parts. The first is the action grammar (a SCFG) that describes the hierarchical structure of all the activities to be recognized. Second is the hierarchical Bayesian network that is generated from the action grammar. Third is the final module that takes a stream of input symbols (level 1 action symbols) and uses deleted interpolation to determine the current probability distribution across each possible output symbol (level 2 action symbol). The details of the proposed system are described here based on the use of the CAVIAR data set[7)] to provide concrete explanation of each aspect of the algorithm.

4.2 Action grammar

The set of all terminal and non-terminal symbols The set of terminals (level 1 action symbols) $\mathbf{T}$, the set of action symbols (called level 2 actions) $\mathbf{A}$ and $\mathbf{I}$ are given in Table 5 and 6). Accordingly, the set of nonterminals $\mathbf{N}$ is defined as $\mathbf{N} = \mathbf{I} \cup \mathbf{A}$. The set of production rules Σ and their corresponding probabilities are given in Table 7. Although the grammar here is manually defined, it is clear from section 3 that the grammar can also be learned.

Table 5 Definition of the level 1 actions (terminal symbols).

Level 1 Actions $\mathbf{T}$	Meaning
en	enter : appears in the scene
ex	exit: disappears from the scene
ne	near exit/ entrance : moving near an exit / entrance
br	browse : standing near landmark
in	inactive: standing still
mp	move in place : standing but moving
wa	walk : moving within a certain velocity range
pd	put down : release object
pu	pick up : contact with object

Table 6 Definition of the level 2 actions and intermediate actions (nonterminal symbols).

Level 2 Actions $\mathbf{A} \in \mathbf{N}$	Meaning
AR	Arriving : Arriving into the scene
BI	Being Idle : Spending extra time in the scene
BR	Browsing : Showing interest in an object in the scene
TK	Taking away : Taking an object away
LB	Leaving behind : Leaving an object behind
PT	Passing Through : Passing through the scene
DP	Departing : Leaving the scene
Intermediate Actions $\mathbf{I} \in \mathbf{N}$	**Meaning**
AI	Action in Place: Taking action while in place
MV	Moving : Moving with a minimum velocity
MT	Move to : Moving in place after walking
MF	Move from : Walking after moving in place

4.2.1 Hierarchical Bayesian network

Despite the expressive power of the SCFG, they were created to characterize formal language and thus in general, syntactic parsers are not well-suited for handling noisy data. Bayesian networks have the robustness needed to deal with faulty sensor data, especially when dealing with human actions. In contrast to standard parsing algorithms, the merit of using an BN is found in the wide range of queries that can be executed over the network[32)]. In addition, BNs can deal with negative evidence, partial observations (likelihood evidence) and even missing evidence, making it a favorable framework for vision applications that deal with uncertain observations.

A previously proposed method[32)] is used to transform the action grammar (level 2 grammar) into a hierarchical Bayesian network (HBN). The term HBN is used here because information about hierarchy from the SCFG is embedded in the BN. By converting the action grammar into a HBN, evidence nodes $\mathbf{E}$

Table 7 Level 2 action grammar.

S → BI	0.20	BR → br	0.20
S → BR	0.10	BR → MV br	0.20
S → TK	0.05	BR → br mp	0.30
S → LB	0.05	BR → MV br mp	0.30
S → PT	0.30		
S → AR	0.15	LB → pd	0.50
S → DP	0.15	LB → MV pd	0.20
		LB → pd mp	0.05
BI → AI	0.10	LB → pd wa	0.05
BI → MV AI	0.10	LB → pd mp wa	0.10
BI → AI MV	0.10	LB → mp pd mp	0.10
BI → mp AI MV	0.10		
BI → mp	0.20	DP → ex	0.40
BI → MF mp	0.10	DP → wa ne ex	0.30
BI → MF	0.10	DP → ne ex	0.20
BI → MV ne MV	0.10	DP → wa ne	0.10
BI → AI wa ne	0.10		
		MV → MF	0.20
TK → pu	0.50	MV → MT	0.20
TK → MV pu	0.20	MV → wa	0.30
TK → pu mp	0.20	MV → mp	0.30
TK → pu wa	0.10		
TK → MV pu MV	0.10	MF → mp wa	1.00
		MT → wa mp	1.00
PT → en wa ex	0.70		
PT → ne wa ne	0.30	AI → in	0.60
		AI → br	0.20
AR → en	0.50	AI → pu	0.10
AR → en MV	0.50	AI → pd	0.10

contain subsets of terminal symbols **T**, query nodes **Q** contain subsets of level 2 actions **A** and hidden nodes **H** contain subsets of indexed production rules **R**. The result of transforming the grammar in Table 7 into a HBN is depicted in Figure 10.

The probability density function (PDF) for level 2 actions is denoted as $p(\mathbf{A}|\mathbf{e})$ where $\mathbf{A} = \{A_1, A_2, \ldots, A_v\}$ is the set of all level 2 actions (states). The input vector $\mathbf{e} = [e_1, e_2, \ldots, e_L]$ is a string of evidence at the evidence nodes of the HBN where L is the maximum length of the HBN. The probability of a specific level 2 action is defined as the sum of the probabilities from each of the query nodes

$$p(A_i|\mathbf{E}) = p(Q_1 = A_i|\mathbf{E}) + \cdots + p(Q_u = A_i|\mathbf{E}). \quad (15)$$

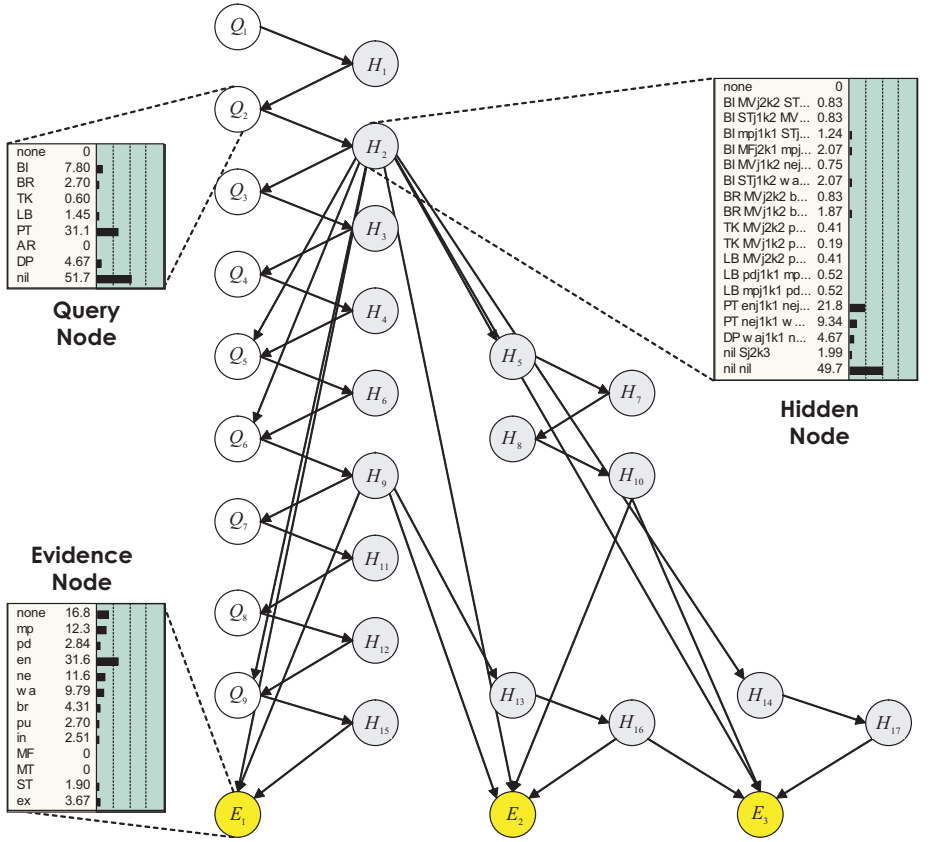

Fig. 10 Hierarchical Bayesian Network (maximum length $l = 3$). The content of each node type is depicted by a bar chart.

4.3 Deleted interpolation

The concept of deleted interpolation (DI) involves combining two (or more) models of which one provides a more precise explanation of the observations but is not always reliable and the another which is more reliable but not as precise. It is called *deleted* interpolation because the models which are being interpolated use a subset of the conditioning information of the most discriminating function[21].

In the proposed system it is assume that the analysis of a long sequence of

evidence is more precise than that of a shorter length because a long sequence takes into consideration more information. However, when analysis over a long (more precise) input sequence fails one would like to fall back on analysis based on a shorter (more reliable) subsequence.

To implement this the current probability distribution $\mathbf{S}$ across level 2 actions, at each time instance, is calculated as a weighted sum of models,

$$\mathbf{S} = \sum_{i=1}^{L} \lambda_i p(\mathbf{A}|\mathbf{O}_i), \tag{16}$$

where $\mathbf{O}_i$ is the string of full evidence when $i = 1$ and represents smaller subsets of the evidence sequence as the index i increases. The weights are constrained by $\sum_{i=1}^{L} \lambda_i = 1$. This is the mechanism that effectively allows the system to represent overlapped activities.

4.4 Experiments

Since the ground truth for each agent in each frame is labeled in XML (information about position, appearance, movement, situation, roles, and context), the ground truth data is used directly as the low-level input into the system for practical reasons. Each video sequence was processed to create a sequence of level 1 action symbols by applying the logic equations to the XML data. It is noted here that as presented in section 2, the actions symbols can also be produced by a probabilistic classifier.

The following experiments show that the proposed method is well-suited for recognizing sequential and overlapped single-agent activities. In the first two experiments it is shown that the use of DI improves performance as opposed to not using DI. In the latter two sections, the effect of the values chosen for grammar rule probabilities and the mixture weights are examined. It is shown that the parameters of the grammar and the parameters of the mixture weight have only a minimal impact on the results.

The video data used for this experiment was taken in a lobby environment (Figure 11) and the sequence of level 1 actions were generated using the labeled CAVIAR data. Analysis was run on six video sequences (Walk1, Walk2, Browse1, Browse2, Leave1 and Leave2) to test the performance of the system. The results for the Leave1 sequence is given here (Figure 12) where the ground truth is

Fig. 11 Key frames for the "Leave Behind and Pick Up" (Leave1) sequence.

given along with the results for each of the four different experimental setups. The ground truth was compiled from multiple users, as a normalized sum of the interpretations of the video data.

The overall precision rate was 88% after filtering out a common problem. An instance of temporal concurrence between activities is depicted in Figure 12(b) between *Being Idle* and three other activities. The recall (capture) rate was 59% (equivalently, a miss rate of 41%) which indicates that the system was not able to detect the activity for the complete duration of the level 2 action as described by the ground truth data. The false alarm rate was 3% (not including the effects of *Passing Through*). The low false alarm rate is expected because the input symbols (level 1 actions) only change when there is a significant change in an

agents visual characteristics.

To understand the advantage of using DI, an experiment was performed again on the same sequences but without the use of DI (Figure 12(b)). Since subsequences of the evidence are not used to interpolate the results, several level 2 actions based on smaller strings were not detected by the system.

To examine the effect of the grammar parameters, the grammar probabilities were set to be uniform. It is interesting to observe that the proportion of the probabilities between activities remain virtually unchanged after rule probabilities have been changed (Figure 12(d)). Likewise the DI weight were also set to be uniform to evaluate the effect of the weights. The results remained similar to the results of using the original weighting scheme (Figure 12(e)). From these two experiments, it is observed that the structural analysis of a symbol sequence plays a larger role in determining the results compared to the role of the probabilities of the rules or the weights of the interpolation.

4.5 Summary

This section has proposed an activity recognition framework robust to overlapped activities based on interpolation between hierarchical activity models. In particular, a stocastic context-free activity grammar was converted into a HBN to allow the system to make complex probabilistic queries. The HBN was then used to discover overlapped activities over a string of discrete primitive action symbols via DI. Through a set of preliminary experiments, it was shown that the proposed methodology is well-suited for detecting the co-occurrence of simple single-agent activities.

5. Conclusion

This paper has presented a bottom-up computational framework for modeling, learning and recognizing human activities. In the first section, it was shown that by describing primitive actions as a combination of both motion and visual context, the proposed algorithm is able to correctly categorize actions from a video database of actions. As a result, the segments of an action sequence were labeled according to the respective class yielding a string of action symbols. In the second section it was shown that by testing various hypothesis using an MDL criterion enabled the proposed system to discover the basic structure of an ac-

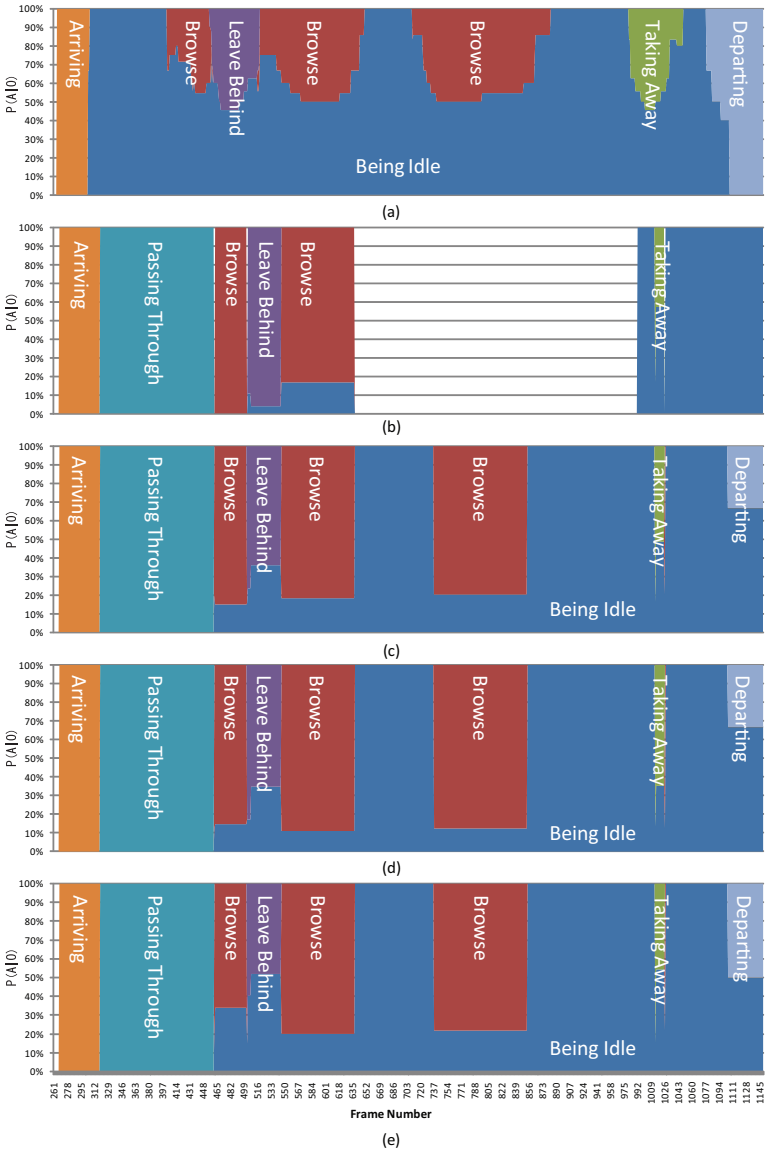

Fig. 12 Leave bag sequence (a) ground truth (b) no DI (c) DI with user defined rule probabilities (d) DI with uniformly distributed rule probabilities (e) DI with uniform mixture weights

tivity sequence from a symbol string of primitive actions corrupted by noise. As a result, an optimal SCFG expressing the grammar of the activities contained in the action string was acquired. In the third section it was shown that given a stochastic context-free grammar that describes human activity, the activities occurring within a stream of observations (a string of action symbols) can be detected, even when the activities are overlapped. Taken as a whole, the algorithms presented in this paper describe a novel prototype system for learning and recognizing human activities from video sequences.

References

1) Ayers, D. and Shah, M.: Monitoring Human Behavior from Video Taken in an Office Environment, *Image Vision Comput.*, Vol.19, No.12, pp.833–846 (2001).
2) Blei, D.M., Ng, A.Y. and Jordan, M.I.: Latent Dirichlet Allocation, *Journal of Machine Learning Research*, Vol.3, pp.993–1022 (2003).
3) Boiman, O. and Irani, M.: Detecting Irregularities in Images and in Video, *Proceedings of the International Conference on Computer Vision*, pp.I:462–469 (2005).
4) Bouguet, J.Y.: Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm (2002).
5) Brand, M.: Understanding Manipulation in Video, *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, p.94 (1996).
6) Bui, H.H., Venkatesh, S. and West, G. A.W.: Tracking and Surveillance in Wide-Area Spatial Environments Using the Abstract Hidden Markov Model, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol.15, No.1, pp. 177–195 (2001).
7) CAVIAR: IST Fifth Framework Programme (IST-2001-37540). Found at http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.
8) Dollár, P., Rabaud, V., Cottrell, G. and Belongie, S.: Behavior Recognition via Sparse Spatio-Temporal Features, *Proceedings of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp.65–72 (2005).
9) Fagg, A.H. and Arbib, M.A.: Modeling parietal–premotor interactions in primate control of grasping, *Neural Networks*, Vol.11, No.7-8, pp.1277–1303 (1998).
10) Fanti, C., Zelnik-Manor, L. and Perona, P.: Hybrid models for human motion recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1166–1173 (2005).
11) Gupta, A. and Davis, L.S.: Objects in Action: An Approach for Combining Action Understanding and Object Perception, *Proceedings of the IEEE Conference on Computer Vision*, pp.1–8 (2007).
12) Hamid, R., Johnson, A.Y., Batta, S., Bobick, A.F., Isbell, C.L. and Coleman, G.: Detection and Explanation of Anomalous Activities: Representing Activities as Bags of Event n-Grams, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.I: 1031–1038 (2005).
13) Hofmann, T.: Probabilistic Latent Semantic Analysis, *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp.289–29 (1999).
14) Ivanov, Y.A. and Bobick, A.F.: Recognition of Visual Activities and Interactions by Stochastic Parsing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.22, No.8, pp.852–872 (2000).
15) Kitani, K.M., , Okabe, T., Sato, Y. and Sugimoto, A.: Discovering Primitive Action Categories by Leveraging Relevant Visual Context, *Proceedings of the IEEE International Workshop on Visual Surveillance*, pp.1–8 (2008).
16) Kitani, K.M., Sato, Y. and Sugimoto, A.: Deleted Interpolation using a Hierarchical Bayesian Grammar Network for Recognizing Human Activity, *Proceedings of the Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp.239–246 (2005).
17) Kitani, K.M., Sato, Y. and Sugimoto, A.: Recovering the Basic Structure of Human Activities from a Video-Based Symbol String, *Proceedings of the IEEE Workshop on Motion and Video Computing*, pp.9–9 (2007).
18) Laptev, I.: On space-time interest points, *International Journal on Computer Vision*, Vol.64, No.2, pp.107–123 (2005).
19) Lee, D.D. and Seung, H.S.: Learning the parts of objects by non-negative matrix factorization, *Nature*, Vol.401, pp.788–791 (1999).
20) Lowe, D.G.: Object Recognition from Local Scale-Invariant Features, *Proceedings of the International Conference on Computer Vision*, p.II:1150 (1999).
21) Manning, C.D. and Schutze, H.: *Foundations of Statistical Natural Language Processing*, The MIT Press (2003).
22) Minnen, D., Essa, I.A. and Starner, T.: Expectation Grammars: Leveraging High-Level Expectations for Activity Recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.II: 626–632 (2003).
23) Moore, D.J. and Essa, I.A.: Recognizing Multitasked Activities from Video Using Stochastic Context-Free Grammar, *Proceedings of the National Conference on Artificial Intelligence*, pp.770–776 (2002).
24) Moore, D.J., Essa, I.A. and Hayes, M.H.: Exploiting Human Actions and Object Context for Recognition Tasks, *Proceedings of the IEEE International Conference on Computer Vision*, pp.80–86 (1999).
25) Nevil-Manning, C.G. and Witten, I.H.: Online and Offline Heuristics for Inferring Hierarchies of Repetitions in Sequences, *Proceedings of IEEE*, 88, No.11, pp.1745–1755 (2000).
26) Nguyen, N.T., Bui, H.H., Venkatesh, S. and West, G. A.W.: Recognising and Monitoring High-Level Behaviours in Complex Spatial Environments, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.II: 620–

625 (2003).
27) Niebles, J.C. and Fei-Fei, L.: A Hierarchical Model of Shape and Appearance for Human Action Classification, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1–8 (2007).
28) Niebles, J.C., Wang, H. and Fei-Fei, L.: Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words, *Proceedings of the British Machine Vision Conference*, pp.III:1249–1258 (2006).
29) Nigam, K., McCallum, A., Thrun, S. and Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, Vol.39, No.2-3, pp.103–134 (2000).
30) Oliver, N., Horvitz, E. and Garg, A.: Layered Representations for Human Activity Recognition, *Proceedings of the IEEE International Conference on Multimodal Interfaces*, IEEE Computer Society, pp.3–8 (2002).
31) Oliver, N.M., Rosario, B. and Pentland, A.: A Bayesian Computer Vision System for Modeling Human Interactions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.22, No.8, pp.831–843 (2000).
32) Pynadath, D.V. and Wellman, M.P.: Generalized Queries on Probabilistic Context-Free Grammars, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, No.1, pp.65–77 (1998).
33) Ryoo, M.S. and Aggarwal, J.K.: Recognition of Composite Human Activities through Context-Free Grammar Based Representation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1709–1718 (2006).
34) Schuldt, C., Laptev, I. and Caputo, B.: Recognizing Human Actions: A Local SVM Approach, *Proceedings of the International Conference on Pattern Recognition*, pp. 32–36 (2004).
35) Shi, J. and Tomasi, C.: Good Features to Track, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (1994).
36) Shi, Y., Huang, Y., Minnen, D., Bobick, A.F. and Essa., I.A.: Propagation Networks for Recognition of Partially Ordered Sequential Action, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.862–869 (2004).
37) Siskind, J.M.: Visual Event Classification via Force Dynamics, *Proceedings of the National Conference on Artificial Intelligence and Conference on Innovative Applications of Artificial Intelligence*, pp.149–155 (2000).
38) Starner, T. and Pentland, A.: Real-time American Sign Language recognition from video using hidden Markov models, *Proceedings of the International Symposium on Computer Vision*, p.265 (1995).
39) Stolcke, A.: Bayesian Learning of Probabilistic Language Models, PhD Thesis, University of California at Berkeley (1994).
40) Teh, Y.W., Jordan, M.I., Beal, M.J. and Blei, D.M.: Hierarchical Dirichlet processes, *Journal of the American Statistical Association*, Vol.101, No.476, pp.1566–1581 (2006).
41) Wada, T. and Matsuyama, T.: Appearance Based Behavior Recognition by Event Driven Selective Attention, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.759–764 (1998).
42) Wang, T., Shum, H., Xu, Y. and Zheng, N.: Unsupervised Analysis of Human Gestures, *Proceedings of the IEEE Pacific Rim Conference on Multimedia*, pp.174–181 (2001).
43) Wang, X., Ma, X. and Grimson, E.: Unsupervised Activity Perception by Hierarchical Bayesian Models, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1–8 (2007).
44) Wong, S., Kim, T. and Cipolla, R.: Learning Motion Categories using both Semantic and Structural Information, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1–6 (2007).
45) Yamato, J., Ohya, J. and Ishii, K.: Recognizing Human Action in Time-Sequential Images using Hidden Markov Model, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, pp.379–385 (1992).
46) Zacks, J.M. and Tversky, B.: Event Structure in Perception and Conception, *Psychological Bulletin*, Vol.127, pp.3–21 (2001).