Multi-thread Parsing for Recognizing Complex Events in Videos

Zhang Zhang, Kaiqi Huang, and Tieniu Tan

National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Science, Beijing 100190, China
{zzhang,kqhuang,tnt}@nlpr.ia.ac.cn

Abstract. This paper presents a probabilistic grammar approach to the recognition of complex events in videos. Firstly, based on the original motion features, a rule induction algorithm is adopted to learn the event rules. Then, a multi-thread parsing (MTP) algorithm is adopted to recognize the complex events involving parallel temporal relation in sub-events, whereas the commonly used parser can only handle the sequential relation. Additionally, a Viterbi-like error recovery strategy is embedded in the parsing process to correct the large time scale errors, such as insertion and deletion errors. Extensive experiments including indoor gymnastic exercises and outdoor traffic events are performed. As supported by experimental results, the MTP algorithm can effectively recognize the complex events due to the strong discriminative representation and the error recovery strategy.

1 Introduction

Recently, event recognition in videos has become one of the most active topics in computer vision. A great deal of researchers have worked on it, which ranged from the recognition of simple, short-term actions, such as running and walking [1], to complex, long-term, multi-agent events, such as operating procedures or multi-agent interactions [3], [5], [11].

In this paper, we focus on the recognition of complex events involving multiple moving objects. Motivated by a natural cognitive experience that a complex event can be treated as a combination of several sub-events, we propose the solution based on some syntactic pattern recognition techniques.

The flowchart of our solution is shown in Fig. 1. The motion features of moving objects are obtained by tracking. Then in the procedures of *Primitive Modeling* and *Event Rule Induction*, we take advantage of the method developed by Zhang et al. [14] to obtain a number of primitives as well as a set of event rules. The learnt rules extend the Stochastic Context Free Grammar (SCFG) production with Allen's temporal logic [17] to represent the complex temporal relation in sub-events. However, in recognition module, the commonly used parser cannot handle the sub-events with parallel temporal relations. To solve this problem, referring to the idea in [18] where the linear ordered constraint in *identifiers(ID) set* is relaxed to an unordered one, we extend the original Earley-Stolcke parser [20]

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part III, LNCS 5304, pp. 738–751, 2008.

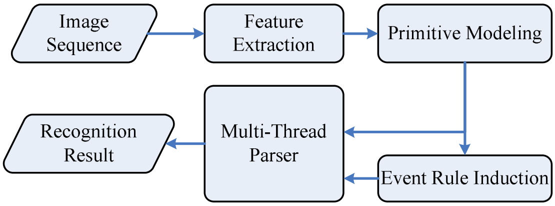


Fig. 1. Flowchart of the solution to complex event recognition in this work

to a multi-thread parsing (MTP) algorithm. Additionally, a Viterbi-like error recovery strategy is also embedded to correct the large scale errors, e.g. insertion and deletion errors in the input stream.

As examples, experiments including gymnastic exercises and traffic events are performed. As supported by experimental results, the effectiveness and robustness of the MTP algorithm has been validated.

2 Related Work

There has been much work on event recognition in videos. For simple actions such as "walking" and "running", some researchers developed a quantity of effective feature descriptors from raw images [1] [2], etc.

For recognizing complex long-term events, most work is based on modeling the moving object tracks. Generally two main kinds of approaches are used: Dynamic Bayesian Network (DBN) based approaches and rule based approaches.

In DBN based approaches [3], [5], [4], [6], some trained stochastic state space models are adopted to represent the inherent structure of complex event. These approaches have the advantage of well studied parameter learning algorithms and the capability to reason with uncertainty. However for the multi-agent interactions involving complex temporal relation, the performance seriously relies on the appropriate model topology that is difficult to be learnt from small training data. In most cases, the predefinition for model topology is needed.

In rule based approaches, some primitives (atomics) are first detected, then complex events are recognized as the combination of several sub-events with certain rules [15], [16]. Due to the convenient representation and the efficient parsing algorithm, SCFG has been adopted in applications, such as video surveillance [10], indoor operating activities [9], [8], [12] and human interaction [11]. However in their work, the event rules were all predefined manually, which is impossible in real applications. Furthermore, only single thread event can be tackled, where the temporal relation between sub-events is just sequential relation.

In fact, the problem on parallel relation has been noticed in some previous work. In [11], based on context-free grammar, S.M.Ryoo et al. also used Allen's temporal logic [17] to represent the parallel relation in complex human interactions. In recognition, the parsing problem was turned into a common constraint satisfaction problem. In [18], a multidimensional parsing algorithm is proposed to handle parallel relation in multimodal human computer interaction, where the linear ordered constraint for combining two constituents is relaxed to an unordered one. However, the above two methods do not consider the large time scale errors in primitive detection such as insertion errors and deletion errors.

In this study, we focus on developing a more effective parsing algorithm which can handle the parallel relation problem and the uncertainties in primitive detection simultaneously for recognizing complex visual event.

3 Multi-thread Parsing

As shown in Fig. 1, there are two inputs to the parser: one is a symbol (primitive) stream, the other is a set of event rules. Here each primitive is represented as a four-tuple $\{type, t_s, t_f, lik\}$, where *type* is the primitive type, t_s and t_f represent *start time* point and *finish time* point respectively, *lik* is the likelihood probability. The primitives are arranged into a stream according to t_f ascendingly.

Referring to [14], the SCFG production is extended by a relation matrix:

$$H \quad \rightarrow \quad \lambda \ \ \{R\} \qquad [p] \tag{1}$$

where R is the temporal relation matrix in which the element r_{ij} denotes the temporal relation between the i_{th} sub-event and the j_{th} one, and p is the conditional probability of the production being chosen, given the event H. Note in our experiments, the size of event rule is two at most, so that the temporal relation matrix can be represented as one element r_{12}.

Then the parsing task is to find out the most possible derivation (parsing tree) T to interpret a primitives stream S. In this work, for the root symbol A, a set of rules G_A is constructed. Therefore in terms of Maximum Likelihood (ML) criterion, the event recognition problem can be described as follows:

$$< A_d, T_d >= \arg \max_{<A,T>} P(S, T|G_A) \tag{2}$$

where A_d is the final decision on the type of complex event, T_d is the corresponding parsing tree, and $P(S, T|G_A)$ is computed as the product of the probabilities of the rules used in the parsing tree.

3.1 Parsing Algorithm

Referring to the idea in [18], we propose the multi-thread parsing algorithm by extending the Earley-Stolcke parser [20], where three operations: *scanning*, *completion* and *prediction* are performed iteratively.

Here, the parsing state in our algorithm is represented as follows:

$$I : X \rightarrow \lambda \cdot Y \mu \quad [v] \tag{3}$$

where I is *ID set* that indicates the constituents in the input primitives, the *dot* marker is the current parsing position, which denotes the sub-events λ have been observed and the next needed symbol is Y, μ is the unobserved string, v

is the Viterbi probability which corresponds the maximum possibility derivation of the state. In addition, the temporal attributes are also recorded in the state.

Different from the state in the Earley-Stolcke parser where the *ID set* must be a set of consecutive primitives, the current *ID set* may contain disconnected identifiers. For example, $I = \{3, 5, 7\}$ means the state is comprised of the $3rd$, $5th$ and $7th$ primitives in the input string. The relaxed *ID set* enables multiple parsing threads to exist simultaneously.

Given the current state set $StateSet(i)$ and the primitive, the following three steps are to be performed.

Scanning. For each primitive, say d, a pre-nonterminal rule $D \rightarrow d$ is added, so that the role of *Scanning* is able to accept the current primitive with the predicted state of the pre-nonterminal rule. And the likelihood of the detected primitive will be multiplied by the Viterbi probability of the predicted state.

Completion. For a completed state in $StateSet(i)$, suppose $I'' : Y \rightarrow \omega \cdot [v'']$ that denotes event Y has been recognized, the state S_j in the last state set $StateSet(i-1)$ will be examined with the following conditions:

- Y is one of the unobserved sub-events of S_j.
- $I'' \cap I_{S_j} = \phi$, where the *ID set* of the completed state I'' is not intersected with that of S_j.
- The relations between Y and the observed sub-events of S_j is consistent with the rule definition. The relation is computed by the fuzzy method in [21].

Then for the state satisfying the above conditions, do another judgment in terms of the position where Y locates at in S_j. If Y is not the first unobserved sub-event (the symbol following the *dot*), the unobserved sub-events that are prior to Y are treated as deletion error candidates that are to be handled in Section 3.2, else S_j can be assumed as $I : X \rightarrow \lambda \cdot Y \mu[v]$, a new state is generated.

$$\begin{cases} I : X \rightarrow \lambda \cdot Y\mu[v] \\ I'' : Y \rightarrow \omega \cdot [v''] \end{cases} \Rightarrow I' : X \rightarrow \lambda Y \cdot \mu[v'] \tag{4}$$

where$I' = I \bigcup I''$ and $v' = vv''$.

In the current state set, if another identical state with the new state has existed, the Viterbi probability v_c of the identical state will be modified as $v_c = \max\{v_c, v'\}$, else the new state will be added into the current state set.

Due to the relaxed *ID set*, there may be too many combinations of different primitives. Therefore, a *beam-width constraint* is adopted to prune the redundant states, where only the first ω states are saved according to the Viterbi probability in an isomorphic state set. Here we define two states are isomorphic, if and only if they share the same rule, the same *dot* position, but different *ID set.*

Prediction. As the next symbol may belong to other parsing thread, in prediction all the uncompleted states in the last state set will be put into the current state set. Note, all the non-terminals will be predicted in initialization step.

3.2 Error Recovery Strategy

Commonly there are three kinds of errors: insertion, deletion, and substitution errors. Insertion errors mean the spurious detection of primitives that do not actually happen. Deletion errors denote the missing detection of primitives actually occurred. Substitution errors mean the misclassification between primitives.

In [10], insertion errors were accepted by the extended *skip* productions, nevertheless the deletion errors cannot be handled by such *skip* productions. In [9], three hypothetical parsing pathes corresponding to the three kinds of errors (insertion, deletion, substitution) were generated as the parsing fails to accept the current primitive. However an error may not lead to the failure in current scanning but in the next iteration.

Here referring to the idea in common parsing [19], a number of error hypotheses will be generated along with the parsing process. Finally, a Viterbi-like backtracking will determine the most possible error distribution. Since a substitution error can be seen as a pair of one insertion error and one deletion error, only insertion and deletion errors are considered in the following.

Insertion Error. Due to the relaxed *ID set* in which the *identifiers* may be disconnected, the insertion errors are tackled naturally. At the end of parsing, for each completed root state $I_f : 0 \rightarrow S \cdot [\upsilon_f]$, the primitives that are not contained in I_f are treated as insertion errors of this derivation. The penalties of insertion errors will be added to the Viterbi probability as follows:

$$\upsilon = \upsilon_f \prod_{i \in I'_f} \rho_i \tag{5}$$

where ρ_i is the penalty factor of the ith insertion error with a low value, I'_f is the complement set of I_f.

Deletion Error. As presented in Section 3.1, deletion error candidates may be generated in *completion* operation. Suppose a sate $I'' : X \rightarrow \lambda \cdot Y_1 Y_2 ... Y_n Y \mu[\upsilon'']$ where $Y_1 Y_2 ... Y_n$ are hypothesized as deletion errors, Alg.1 is performed to transform the old state into a new one $I_e : X \rightarrow \lambda Y_1 Y_2 ... Y_n \cdot Y \mu[\upsilon_e]$. Here $An\_s = I'' : X \rightarrow \lambda \cdot Y_1 Y_2 ... Y_n Y \mu[\upsilon'']$, $I' = I \cup I''$, *e_position* is the position where Y locates at in An_s, *s_set* is the last state set.

Concretely, given $Y_i = An\_s.predict$ that is the symbol just behind the *dot* of An_s, if Y_i can only be completed by pre-nonterminal rule $Y_i \rightarrow z$, the terminal z is recovered and a completed state *re_s* is generated by *scanning* operation. The z is assigned to a low likelihood as the penalty factor of deletion error.

Else if Y_i is a non-terminal, Max_Ex is performed to find out the state $re\_s = Y_i \rightarrow \lambda' \cdot Z\mu'$ that is to complete Y_i with maximum Viterbi probability in *s_set*. Then *Recovery* and Max_Ex are performed repeatly, until *re_s* becomes a completed state.

Then *An_s* is combined with *re_s* to form a new state *new_s* with *completion* operator. Finally we examine whether the dot position of *new_s* reaches to *e_position*, if true the recovery of An_s is over, else recover the next sub-event.

Algorithm 1. Recovery(An_s,I',$e\_postion$,s_set)

```
if $An\_s.predict$ is pre-nonterminal then
    $z$ = Error_Hypothesize($An\_s.predict$);
    $re\_s$ = scanning($An\_s$, $z$);
else
    $re\_s$ = Max_Ex($An\_s.predict$, $I'$, $s\_set$);
    while $re\_s.dot < size(re\_s.rule)$ do
        Recovery($re\_s$, $I' \cup I_{re\_s}$, $re\_s.dot + 1$, $s\_set$);
        $re\_s$ = Max_Ex($An\_s.predict$, $I'$, $s\_set$);
    end while
end if
$new\_s$ = completion($An\_s$, $re\_s$);
if $new\_s.dot < e\_position$ then
    Recovery($new\_s$, $I' \cup I_{new\_s}$, $e\_position$, $s\_set$);
else
    Return;
end if
```

Here considering the computing cost, we assume that deletion errors just take a small proportion in a state. Thus a *maximum error constraint* is proposed to prune the states with too many error hypotheses. An exponential distribution is used to model the number of deletion errors. It is written as $e^{-\theta n^{\frac{1}{2}}}$, where θ is a control parameter, n is the size of *ID set* of the state. For a given state with m deletion errors, if $\frac{m}{n} > e^{-\theta n^{\frac{1}{2}}}$, the state will be pruned.

4 Experimental Results

4.1 Gymnastic Exercises

First, a recognizing process for a person doing gymnastic exercises is presented. Three exercises called $E1$, $E2$ and $E3$ are selected from a set of broadcast gymnastics. Twenty nine sequences are collected. The numbers of sequences are 9, 10, and 10 respectively. Fig. 2 illustrates the routine of exercise *E 3*.

Here the motion trajectories of hands and feet are extracted as the original feature. Referring to [7], we use the optical flow magnitude to capture the dominate motion regions. Then hands and feet are located with prior color and spatial information. Note, the tracking technique is not the focus of this paper.

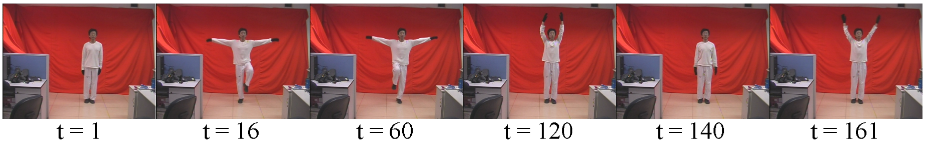
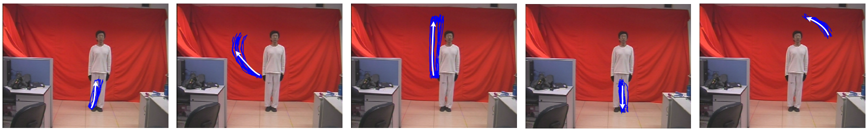
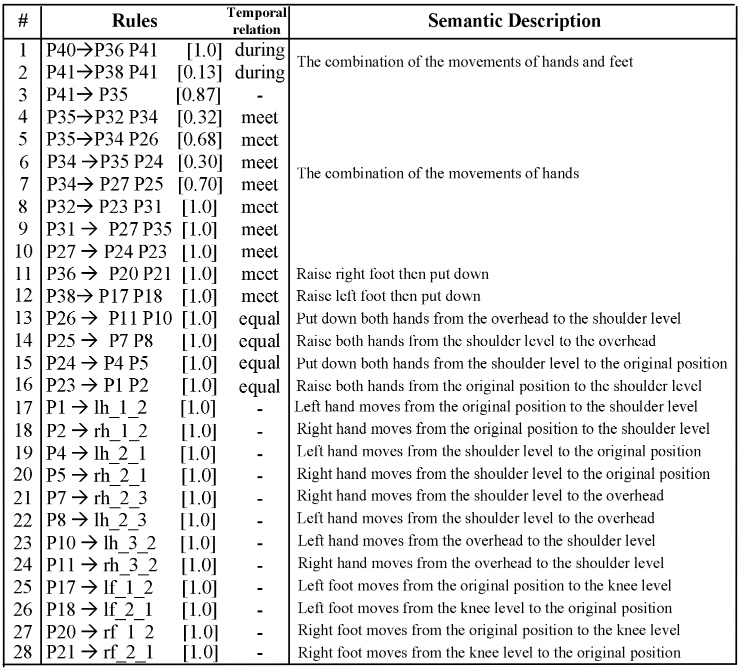


Fig. 2. Illustration on the gymnastic exercises *E 3*



(a) lf_1_2 (b) lh_1_2 (c) lh_1_3 (d) rf_2_1 (e) rh_2_3

Fig. 3. Some of the primitives in the gymnastic exercises. Each primitive describes the movement between different semantic points. For instance, *lh_1_2* means move left hand from semantic point #1 to #2.



#	Rules		Temporal relation	Semantic Description
1	P40→P36 P41	[1.0]	during	The combination of the movements of hands and feet
2	P41→P38 P41	[0.13]	during	
3	P41→ P35	[0.87]	-	The combination of the movements of hands
4	P35→P32 P34	[0.32]	meet	
5	P35→P34 P26	[0.68]	meet	
6	P34 →P35 P24	[0.30]	meet	
7	P34→ P27 P25	[0.70]	meet	
8	P32→ P23 P31	[1.0]	meet	
9	P31 → P27 P35	[1.0]	meet	
10	P27 → P24 P23	[1.0]	meet	
11	P36 → P20 P21	[1.0]	meet	Raise right foot then put down
12	P38→ P17 P18	[1.0]	meet	Raise left foot then put down
13	P26 → P11 P10	[1.0]	equal	Put down both hands from the overhead to the shoulder level
14	P25 → P7 P8	[1.0]	equal	Raise both hands from the shoulder level to the overhead
15	P24 → P4 P5	[1.0]	equal	Put down both hands from the shoulder level to the original position
16	P23 → P1 P2	[1.0]	equal	Raise both hands from the original position to the shoulder level
17	P1 → lh_1_2	[1.0]	-	Left hand moves from the original position to the shoulder level
18	P2 → rh_1_2	[1.0]	-	Right hand moves from the original position to the shoulder level
19	P4 → lh_2_1	[1.0]	-	Left hand moves from the shoulder level to the original position
20	P5 → rh_2_1	[1.0]	-	Right hand moves from the shoulder level to the original position
21	P7 → rh_2_3	[1.0]	-	Right hand moves from the shoulder level to the overhead
22	P8 → lh_2_3	[1.0]	-	Left hand moves from the shoulder level to the overhead
23	P10 → lh_3_2	[1.0]	-	Left hand moves from the overhead to the shoulder level
24	P11 → rh_3_2	[1.0]	-	Right hand moves from the overhead to the shoulder level
25	P17 → lf_1_2	[1.0]	-	Left foot moves from the original position to the knee level
26	P18 → lf_2_1	[1.0]	-	Left foot moves from the knee level to the original position
27	P20 → rf 1 2	[1.0]	-	Right foot moves from the original position to the knee level
28	P21 → rf_2_1	[1.0]	-	Right foot moves from the knee level to the original position

Fig. 4. The learnt rules of the gymnastic exercise *E 3*

For primitive modeling, some semantic points can be firstly learnt by clustering the stop points, since the finish of a basic movement is commonly indicated by the stop motion of hand or foot.

Then the primitives can be considered as the movements between different semantic points. Here, eighteen primitives are obtained. Fig. 3 illustrates some examples. Finally, the trajectory segments belonging to the same primitive are modeled by HMM which is for computing the likelihood of the detected primitive. After primitive detection, each exercise includes around 23 primitives.

For each exercise, a set of rules is learnt by the rule induction algorithm [14]. Fig. 4 describes an example of the learnt rule corresponding to the exercise $E3$, where $E3$ is denoted by non-terminal $P40$. More details of the rule induction algorithm can be found in [14].

To validate the performance of event recognition, HMM and Coupled Hidden Markov Model (CHMM) are chosen for comparison, because they can be trained with little human interference. While other DBN based methods usually require manual construction of model topology. For HMM, the input is a 8-D vector sequence formed by the four trajectories of hands and feet. In CHMM, the four trajectories are divided as two parts for the input of each chain. Here, we take the first 5 sequences for learning rules or parameters and all sequences for test. And the control parameter ω in the *beam-width constraint* is 3. The experimental results are shown in Table 1.

Table 1. Correct classification rate (CCR) on the gymnastic exercises recognition

Event	Truth	MTP			HMM	CHMM
		$\theta = 0.7$	$\theta = 0.5$	$\theta = 0.2$		
E 1	9	9	9	9	9	9
E 2	10	4	10	10	10	10
E 3	10	10	10	10	8	9
Total	29	23	29	29	27	28
CCR		79.3%	100%	100%	93.1%	96.6%

As shown in Table 1, as θ is less than 0.5, the multi-thread parsing (MTP) can recognize all the sequences correctly, whereas HMM misclassifies two sequences and CHMM misclassifies one.

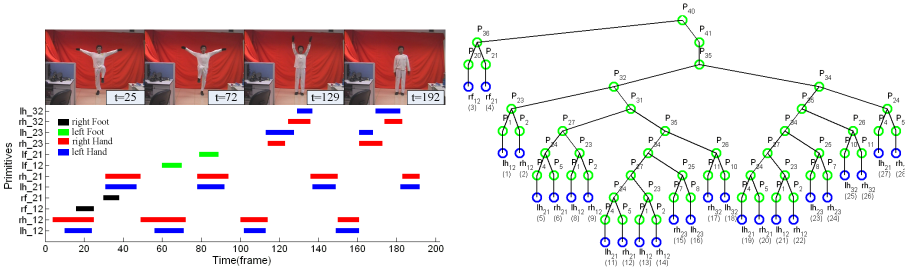
To further validate the robustness of our algorithm, three kinds of synthetic errors are randomly added into the testing trajectories as follows:

- A deletion error is added by replacing a motion trajectory segment that corresponds to a primitive with a still trajectory that does not correspond to any primitive.
- An insertion error is added by replacing a still trajectory segment with a motion trajectory segment that corresponds to a random primitive.
- A substitution error is added by replacing a motion trajectory segment with another segment that corresponds to a different primitive.

After various amounts of large time scale errors are added, we compare our MTP parser and HMM as well as CHMM classifiers again. The performance is shown in Table 2. As six additional errors are added (one substitution error is equivalent to a pair of one insertion error and one deletion error, so there are over 25% errors in the primitive stream), as θ is 0.2 the multi-thread parsing can still acquire a satisfying result 96.6% due to the strong discriminative rule representation and the effective error recovery method. While the performance of HMM and CHMM decreases obviously as the number of errors increases. As θ is 0.5, in terms of the *maximum errors constraint* in Section 3.2, the maximum tolerant number of the deleted errors is $23 * \exp(-1 * 0.5 * \sqrt{23}) \approx 2$. So the performance will decrease rapidly when the number of errors exceeds 2.

Table 2. CCRs on event recognition with synthetic errors

Number of Errors	MTP		HMM	CHMM
	$\theta = 0.5$	$\theta = 0.2$		
1	93.1%	100%	86.2%	89.7%
2	82.8%	100%	82.8%	89.7%
3	72.4%	100%	75.9%	86.2%
4	62.1%	100%	69%	79.3%
5	55.2%	100%	55.2%	75.9%
6	41.4%	96.6%	51.7%	75.9%



(a) primitive detection (b) parsing tree

Fig. 5. An example on the recognition process with our methods. The exercise “E 3” is recognized correctly. Here the leaf nodes are primitives. The number under the primitive indicates the corresponding *ID* in primitive stream.

From the above comparison, the effectiveness and robustness of our methods are validated. Moveover, along with the parsing, a parsing tree can be obtained to express the hierarchical structure explicitly in each primitive stream. An example on whole parsing process is shown in Fig. 5.

In terms of the parsing tree, each input primitive has two possible afflictions. One is that it is accepted by the parsing tree. The other is the identification as an insertion error. Thereafter, the metric *overall correct rate (OCR)* is adopted to measure the parsing accuracy, which can be defined as $\frac{NA+NI}{NP}$, where NA is the number of correct acceptance of primitives in the parsing tree and NI is the number of correct detection of insertion errors, NP is the total number of primitives in input stream.

Table 3 presents the parsing accuracy with original data as well as various additional errors, where ω is 3 and θ is 0.2. As shown in the table, most correct

Table 3. Parsing accuracy in recognizing the gymnastic exercises. Here, #e=0 means the ordinal data, #e=1 is the data with one synthetic errors, and so on.

Errors number	#e = 0	#e = 1	#e = 2	#e = 3	#e = 4	#e = 5	#e = 6
OCR	87.6%	86.1%	85.9%	83.3%	81.3%	81.2%	76.8%

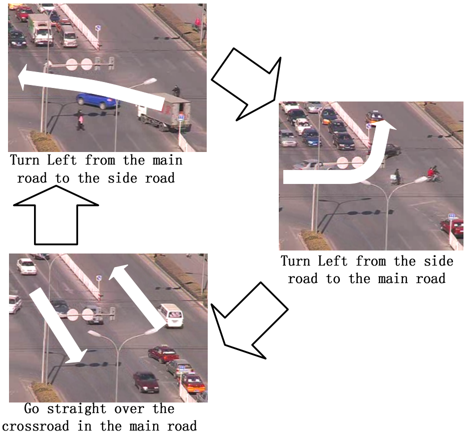

Turn Left from the main road to the side road
Turn Left from the side road to the main road
Go straight over the crossroad in the main road

Fig. 6. Illustration on the traffic events in the crossroad. The trajectory stream in the scene can be represented by an iteration of three kinds of passing event.

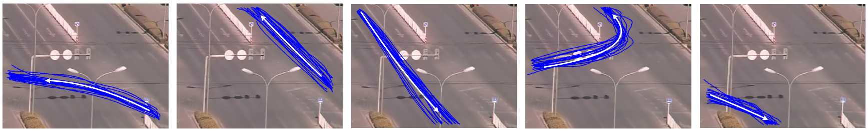
primitives are accepted by the parsing tree, while the parsing accuracy decreases with the increase of the added errors. The failure to accept the primitive is mainly due to two reasons. One is the uncertainty in computing the temporal relation between sub-events where the fuzzy method [21] relies on an appropriate threshold. The other reason is that in the deletion error recovery, only the state with the maximum Viterbi probability is handled, which may lead to the local optimization, instead of the global optimization.

4.2 Traffic Events in Crossroads

To further validate the effectiveness of our method, we would like to test it in a realistic surveillance scene which is shown in Figure 6. In this scene, a traffic cycle composes of three sub-events "*Go straight over the crossroad in the main road*", "*Turn left from the main road to the side road*" and "*Turn left from the side road to the main road*", which happen alternately. Furthermore, "*Go straight over the crossroad in the main road*" can be decomposed into two parallel sub-events "*Go straight over the crossroad in the left side of the main road*" and "*Go straight over the crossroad in the right side of the main road*". Eventually, each traffic event is comprised of a number of primitives which are represented as vehicle trajectories in different lanes.

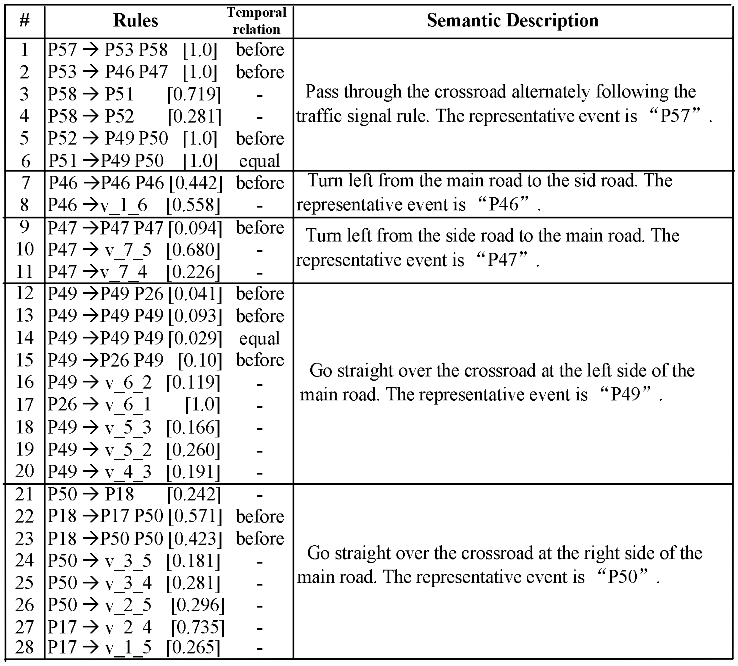
We obtain the trajectory data from the previous work by Zhang et al. [14] that focuses on learning the rules from trajectory stream. In this study, we validate the effectiveness of the MTP parser to recognize the events in trajectory stream.

Here, the single vehicle passing through the scene is considered as primitive. By clustering, seventeen primitives are acquired, which describe the main motion patterns between different entries and exits in the scene. The entries and exits can be learnt by some work on semantic scene modeling such as [13]. Some of the primitives are presented in Fig. 7. In terms of these clusters, the trajectories that do not belong to any of the clusters are deleted. Furthermore as reported



(a) v_1_6 (b) v_2_5 (c) v_5_2 (d) v_7_5 (e) v_8_1

Fig. 7. The basic motion patterns in the crossroad scene. The white arrow denotes the motion direction of moving object.



#	Rules	Temporal relation	Semantic Description
1	P57 → P53 P58 [1.0]	before	Pass through the crossroad alternately following the traffic signal rule. The representative event is "P57" .
2	P53 → P46 P47 [1.0]	before	
3	P58 → P51 [0.719]	-	
4	P58 → P52 [0.281]	-	
5	P52 → P49 P50 [1.0]	before	
6	P51 →P49 P50 [1.0]	equal	
7	P46 →P46 P46 [0.442]	before	Turn left from the main road to the sid road. The representative event is "P46" .
8	P46 →v_1_6 [0.558]	-	
9	P47 →P47 P47 [0.094]	before	Turn left from the side road to the main road. The representative event is "P47" .
10	P47 → v_7_5 [0.680]	-	
11	P47 →v_7_4 [0.226]	-	
12	P49 →P49 P26 [0.041]	before	Go straight over the crossroad at the left side of the main road. The representative event is "P49" .
13	P49 →P49 P49 [0.093]	before	
14	P49 →P49 P49 [0.029]	equal	
15	P49 →P26 P49 [0.10]	before	
16	P49 → v_6_2 [0.119]	-	
17	P26 → v_6_1 [1.0]	-	
18	P49 → v_5_3 [0.166]	-	
19	P49 → v_5_2 [0.260]	-	
20	P49 → v_4_3 [0.191]	-	
21	P50 → P18 [0.242]	-	Go straight over the crossroad at the right side of the main road. The representative event is "P50" .
22	P18 →P17 P50 [0.571]	before	
23	P18 →P50 P50 [0.423]	before	
24	P50 → v_3_5 [0.181]	-	
25	P50 → v_3_4 [0.281]	-	
26	P50 → v_2_5 [0.296]	-	
27	P17 → v 2 4 [0.735]	-	
28	P17 → v_1_5 [0.265]	-	

Fig. 8. The learnt rules in traffic event experiment. v_i_j is the primitive which indicates the basic motion pattern of "moving from the ith entry to the jth exit in the scene".

in [14], some irrelevant trajectories which are unrelated to the traffic rules will distort the rule induction process. Therefore we manually delete these unrelated trajectories, such as v_8_1 in Fig. 7.

The learnt rules are shown in Fig. 8. We find that four main traffic events "*P46*", "*P47*", "*P49*" and "*P50*" (the meanings can refer to Fig. 8) in the crossroad have been learnt. And the whole traffic cycle is denoted by "*P57*". With the learnt rules, the MTP parser is adopted to recognize the interesting events in a given primitive stream.

Twenty traffic cycles are used for testing. Among these cycles, five of them are lack of main sub-event "*P46*" or "*P47*", since there is no vehicles passing in the corresponding ways. Twenty traffic cycles are all recognized correctly, as the

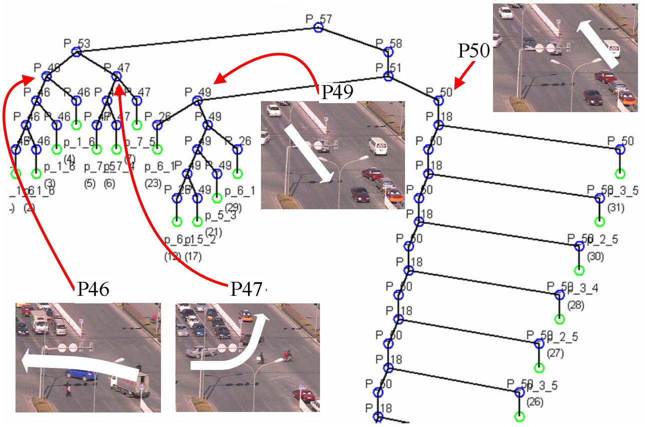


Fig. 9. An example of the parsing result in one traffic cycle. Four main sub-events are all recovered correctly.

Table 4. Parsing accuracy on recognizing traffic events

Event	P46	P47	P49	P50	P57
OCR	100%	100%	96.9%	99.2%	98.9%

root "*P57*" can be recovered in each parsing tree. And the absent sub-events are recovered as deletion errors. An example of the parsing tree is shown in Fig. 9.

Different from the previous gymnastic exercises, we do not use the DBN based method for comparison. That is because the number of moving objects in one frame or an uniform time interval is not a fixed value so that the feature dimensionality cannot be determined.

Moreover, in each parsing tree, we examine the parsing accuracies of the whole traffic cycle "*P57*" as well as four main sub-events "*P46*", "*P47*", "*P49*" and "*P50*". *OCR* presented in Section 4.1 is used to measure the parsing accuracy. Table 4 presents the experimental results. The high *OCR* validates the event rules' capability to fit the primitive stream as well as the accuracy of our parsing algorithm.

5 Conclusion and Future Work

We have present a probabilistic grammar approach to the recognition of complex events in videos. Compared with previous grammar based work, our work has three main advantages:

- In this work, the event rules are learnt by an rule induction algorithm, while in other work the rules are predefined manually.
- The complex event containing parallel sub-events can be recognized by the MTP parser, while others can only handle the single thread event.

– An effective error recovery strategy is proposed to enhance the robustness of the parsing algorithm.

Extensive experiments including indoor gymnastic exercises and outdoor traffic events have been performed to validate the proposed method.

In the future, we will adopt some probabilistic methods to compute the temporal relation. Furthermore, a more efficient parsing strategy is also needed to reduce the time cost.

Acknowledgement

This work is supported by the National Basic Research Program of China (Grant No. 2004CB318110), the National Natural Science Foundation of China (Grant No. 60723005, 60605014, 60332010, 60335010 and 2004DFA06900), the CASIA Innovation Fund for Young Scientists. The authors also thank to Shiquan Wang and Liangsheng Wang for their helps on motion tracking.

References

1. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised Learning of Human Action Categories Using Spatial-TemporalWords. In: Proc. Conf. BMVC (2006)
2. Laptev, I., Lindeberg, T.: Space-time interest points. In: Proc. Int. Conf. on Computer Vision (ICCV) (2003)
3. Laxton, B., Lim, J., Kriegman, D.: Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2007)
4. Shi, Y., Huang, Y., Minnen, D., Bobick, A., Essa, I.: Propagation networks for recognition of partially ordered sequential action. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2004)
5. Nguyen, N.T., Phung, D.Q., Venkatesh, S., Bui, H.: Learning and Detecting Activities from Movement Trajectories Using the Hierarchical Hidden Markov Model. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2005)
6. Xiang, T., Gong, S.: Beyond Tracking: Modelling Activity and Understanding Behaviour. International Journal of Computer Vision (IJCV) 67(1) (2006)
7. Min, J., Kasturi, R.: Activity Recognition Based on Multiple Motion Trajectories. In: Proc. Int. Conf. on Pattern Recognition (ICPR), pp. 199–202 (2004)
8. Minnen, D., Essa, I., Starner, T.: Expectation Grammars: Leveraging High-Level Expectations for Activity Recognition. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 626–632 (2003)
9. Moore, D., Essa, I.: Recognizing Multitasked Activities from Video Using Stochastic Context-Free Grammar. In: Proc. Conf. AAAI (2002)
10. Ivanov, Y.A., Bobick, A.F.: Recognition of visual activities and interactions by stochastic parsing. IEEE TRANS. PAMI 22(8), 852–872 (2000)
11. Ryoo, M.S., Aggarwal, J.K.: Recognition of Composite Human Activities through Context-Free Grammar Based Representation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2006)

12. Yamamoto, M., Mitomi, H., Fujiwara, F., Sato, T.: Bayesian Classification of Task-Oriented Actions Based on Stochastic Context-Free Grammar. In: Proc. Int. Conf. on Automatic Face and Gesture Recognition (FGR) (2006)
13. Wang, X., Tieu, K., Grimson, E.: Learning Semantic Scene Models by Trajectory Analysis. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 110–123. Springer, Heidelberg (2006)
14. Zhang, Z., Huang, K.Q., Tan, T.N., Wang, L.S.: Trajectory Series Analysis based Event Rule Induction for Visual Surveillance. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2007)
15. Hakeem, A., Shah, M.: Learning, detection and representation of multi-agent events in videos. Artif. Intell. 171(8-9), 586–605 (2007)
16. Nevatia, R., Zhao, T., Hongeng, S.: Hierarchical Language-based Representation of Events in Video Streams. In: Proc. CVPRW on Event Mining (2003)
17. Allen, J.F., Ferguson, F.: Actions and Events in Interval Temporal Logical. J. Logic and Computation 4(5), 531–579 (1994)
18. Johnston, M.: Unification-based Multimodal Parsing. In: Proc. Conf. on COLING-ACL, pp. 624–630 (1998)
19. Amengual, J.C., Vidal, E.: Efficient Error-Correcting Viterbi Parsing. IEEE TRANS PAMI 20(10), 1109–1116 (1998)
20. Stolcke, A.: An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. Computational Linguistics 21(2), 165–201 (1995)
21. Snoek, C.G.M., Worring, M.: Multimedia event-based video indexing using time intervals. IEEE TRANS Multimedia 7(4), 638–647 (2005)