RECOGNITION OF ACTIONS IN MEETING VIDEOS USING TIMED-AUTOMATA

Lucjan Pelc, Bogdan Kwolek
Rzeszów University of Technology, W. Pola 2, 35-959 Rzeszów, Poland
{lpelc,bkwolek}@prz.rzeszow.pl

Abstract This paper addresses the problem of action recognition in meeting videos. A declarative knowledge provided graphically by the user together with person positions extracted by a tracking algorithm are used to generate the data for recognition. The actions have been formally specified using timed-automata. The specification was verified on the basis of simulation tests as well as an analysis.

Keywords: Action recognition, vision-based people tracking, timed-automata

1. Introduction

Meeting videos are important documents consisting of captured meetings in specialized smart rooms. Research activities concentrate on recording, representing and browsing of meeting videos. Indexing videos on the basis of visual content is a challenging and multidisciplinary task. By using videos with meetings it is possible to recognize selected actions of participants which have been done throughout the meeting. A visual tracking algorithm [3][9] can be utilized to generate the head trajectories of persons attending the meeting. Each of trajectories contains a sequence of successive head positions of the same person. Using such trajectories we can recognize specific actions which have been performed at specific locations. This can be achieved because the location of many elements in the meeting room occupies fixed places and people attending meetings usually follow specific trajectories [8]. Thus some declarative knowledge that had been provided graphically by the user together with person positions extracted from the trajectory can be further utilized by automata to recognize selected actions. To identify the person actions using automata the time conditions should be employed.

A new approach we present in this paper is designated a recognition of selected actions. They have been formally specified using timed-automata TA [1][5]. Such specification was verified on the basis of simulation tests as well as an analysis consisting in verification of conformity with the conditions specified as the temporal logic formulas.

2. Graphical modeling of the meeting events

In each meeting room there are typical locations, where the participants perform activities. The visual structure of the images varies very little over multiple meetings. The recognition of actions can be realized using absolute and relative positions between objects and heads.

The input data for the automata are generated on the basis of rectangular zones specified with a graphical interface [8]. The zones are used to define the specific meeting events at particular places, see Fig. 1. During action recognition the tracking algorithm [7] generates a sequence of successive head positions of the same person whereas the interface indicates the presence of the heads in the zones. Thanks to keeping the consecutive positions of each person, the module, through indicating the events, provides the states for the automata as well as provides information needed for specification of the automata edges.

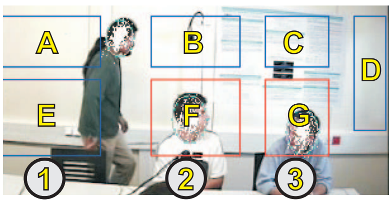

Figure 1. Smart meeting room with rectangular zones of events

The experiments described in this paper have been realized on PETS-ICVS data sets. The PETS data set contains several meeting videos. For cameras 1 and 2 in scenario C there are maximum of 3 people sitting in front of each camera. The goal of the automata is to recognize two actions for each person: work and pause in the work. Seven additional actions are identified and each person is assigned a separate automaton. The relationship between the zones and the automaton states is given in Tab. 1.

Table 1. The relationship between the zones and the events

Zone	State	Event
A	Desk1	standing at the desk #1
B	Desk2	standing at the desk #2
C	Desk3	standing at the desk #3
D	Out	pause in work
E	Char1	sitting at the desk #1
F	Char2	sitting at the desk #2
G	Char3	sitting at the desk #3

3. Timed automata

A timed automaton [1] is basically a finite automaton. It can be perceived as a graph, which contains a finite set of nodes or locations and a finite set of labeled edges, and which is extended with real-valued variables. Such a labeled transition system can be regarded as an abstract model of a timed system. The nonnegative real numbers are used as the time domain. The variables model the logical clocks in the system. The clocks are initialized with zero when the system is initialized and then all clocks increase synchronously with an identical rate. Clock constraints that are guards on edges are employed to achieve the admissible executions and thus the desirable behavior of the automaton. Timed automata contain trajectories, which describe the evolution of the system state.

Each node of the automaton is a state. In timed automata the transitions between states are managed not only by Boolean functions defined over inputs, but clocks too. When a transition is taken, the system performs actions on the clocks and the signals of the system. The clock may be set to zero when a transition is taken.

4. Action modeling and recognition using timed-automata

The timed-automaton that was utilized in this work is depicted in Fig. 2. It contains the states that are given in Tab. 1. as well as the following states:

- `Work1`, `Work2`, `Work3` - indicating that the person works at the desk 1, 2 or 3, respectively
- `Pause` - this state indicates pause in work
- `Move1`, `Move2`, `Move3` - indicating an absence of person in the specified zones. On the basis of the former observations the automaton assumes that the considered person is moving between zones.

We can notice in Fig. 2 that the automaton makes a note for the person entry and the person exit in/from zones corresponding to rectangles in Fig. 1. For example, the entry event into the zone `A`, see Fig. 1, and the exit event from the considered zone, correspond to the automata edges `dk1_on` and `dk1_off`, respectively. The edges carry out the automata to/from the state `Desk1`. By analogy, for the zone `B` it would be `dk2_on` and `dk2_off`, while for the zone `E` corresponding to `Chair1` in the automata, it would be `ch1_on` and `ch1_off`. For the PETS data sets we utilize 3 automata and each one is responsible for an analysis of the behavior of one person.

The automaton recognizes the action `Work` when the considered person is sitting at the desk for a period of time that is larger than the assumed value `Tw`. That means, that the observed person must be in one of the states `E`, `F`, `G`, see Fig. 1 and Tab. 1.

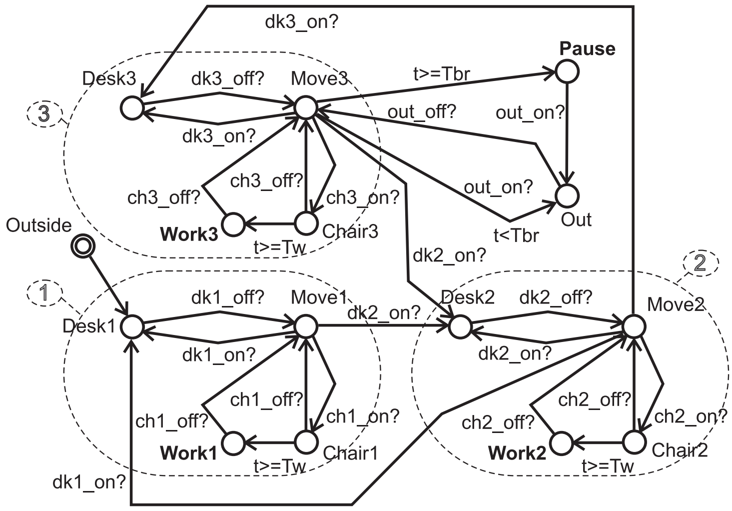

Figure 2. Timed automaton for action recognition

The automaton recognizes the pause in the work when the zone D has been cross-walked as well as the person has not been detected in one of the neighboring zones C or G during the period of time above the threshold Tbr. The automaton can recognize different realizations of particular actions. For example, a person sitting at the desk #3 can first draw oneself up and then exit or can straighten up during exiting. In the first scenario the following sequence of states is registered: Chair3, Move3, Desk3, Out, whereas in the second one the following states are registered: Chair3, Move3, Out, see also Fig. 2.

5. Experiments

Figure 3. depicts the data that have been extracted from the event file. The considered Person1 entered the meeting, has been detected, see also the state Ident in Fig. 3, sat down, worked, and finally exited to make a pause. The urgent states, which are marked with the symbol u placed inside the circle, represent the fact that the person moved without stopping in distinguished states. The situation, where the person was sitting in the space of time Tsit is expressed through state invariant tm<=Tsit and guard of edge tm>=Tsit, see 5th state from top in Fig 3a. Figure 3b. demonstrates some parts of the identification test for Person1. The column Person1 is consistent with Fig. 3a, whereas the column Observer1 demonstrates the action and the states that have been observed during the test. In other words, the column observer is a path of observer automaton from Fig. 2, which corresponds to behavior of Person1. The action Work has been identified successfully.

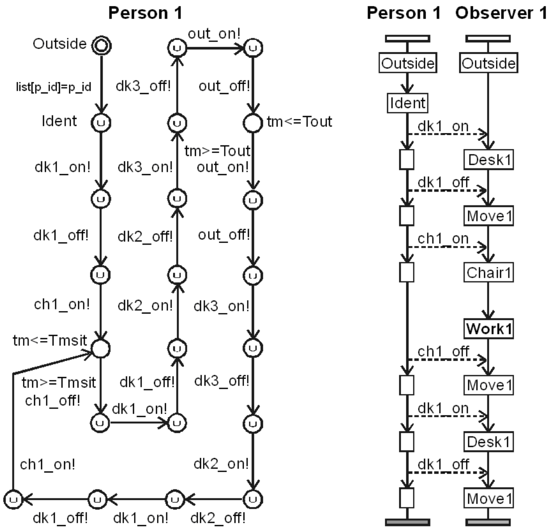

Figure 3. The history of events for `Person1` (a), simulation of identification of action for `Person1` (b)

The tests consisted in simultaneous observation of three persons. For each person a separate automaton was activated. The discussed system does not require that the persons should take the same seat. For example, `Person1` at the beginning the meeting can take seat #1, and after the return to the meeting room he/she can take seat #2. This can take place because the automaton of each person considers all three seats.

Besides the experimental tests a formal verification was done. The correctness conditions have been prepared and verified. The terms have been written as paths formulas of the CTL logic. In Tab. 2., there are some of the gathered terms, which are connected with `Person1`. Symbol `A[]` means: forever, whereas `E<>` means: when the time comes, and the notation Object1.Work1 means: state `Work1` in automaton of `Person1`. The first term in Tab. 2. proofs the system in terms of deadlocks. The second and third are classical reachability properties. They express that the automaton is able to identify the work as well as pause in work of the `Person1`. The last term states the fact that the route from the seat #1 leads in turn through the positions #1, #2 and #3, see Fig. 1 and Fig. 2. The tool Uppaal [2] was utilized in our experiments and tests. The elaborated automata can be utilized in recognition of actions both on-line and off-line.

Particle filtering [4], which is known as CONDENSATION [6] in the area of computer vision, provides a statistical framework of object tracking. Particle filtering has attracted recently much attention due to its great tracking performance in clutter [6][9]. A particle filter built on color and ellipse fitting was utilized to extract the head location for each person in consecutive frames [7]. A background subtraction technique was utilized to detect a person entry/exit into meeting [8].

Table 2. Correctness conditions

Uppal Notation	Meaning
A[] not deadlock	no deadlock
E<> Object.Work1	the automaton is able to identify work
E<> Object.Pause	identification of pause
E<> Object.Desk1 imply Object1.Desk2 imply Object1.Desk3 imply Object.Out imply Object1.Pause	the route from the seat #1 leads in turn through the remaining highlighted positions

6. Conclusions

This paper presents a new approach to recognition of human actions. The proposed approach is based on timed automata. The specification of actions was verified on the basis of simulation tests and analysis. The automata recognize selected human actions in meeting videos. A particle filter based tracking algorithm was used to extract trajectories of persons attending a meeting.

References

[1] Alur R. and D. L. Dill (1994). A Theory of Timed Automata, Theoretical Computer Science, vol. 126, no. 2, 183–235.

[2] Behrmann G., A. David and K. G. Larsen (2004). A Tutorial on Uppal, Aalborg University.

[3] Comaniciu D., V. Ramesh and P. Meer (2000). Real-time Tracking of Non-rigid Objects Using Mean Shift, In Proc. of IEEE Conf. on Comp. Vision and Pat. Rec., 142–149.

[4] Doucet A., S. Godsill and Ch. Andrieu (2000). On Sequential Monte Carlo Sampling Methods for Bayesian Filtering, *Statistics and Computing*, vol. 10, 197–208.

[5] Heitmeyer C. and D. Mandrioli (1996). Formal Methods for Real-Time Computing, Wiley.

[6] Isard M. and A. Blake (1996). Contour Tracking by Stochastic Propagation of Conditional Density, European Conference on Computer Vision, Cambridge, 343–356.

[7] Kwolek B. (2004). Stereovision-based Head Tracking Using Color and Ellipse Fitting in a Particle Filter, European Conference on Computer Vision, LNCS, (4):192–204.

[8] Kwolek B. (2005). Action Recognition in Meeting Videos Using Head Trajectories and Fuzzy Color Histogram, Informatica, (29):281–289.

[9] Perez P., C. Hue, J. Vermaak and M. Gangnet (2002). Color-based Probabilistic Tracking, European Conference on Computer Vision, 661–675.