MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan

14-3

HHMM Based Recognition of Human Activity from Motion Trajectories in Image Sequences

Daiki Kawanaka, Shun Ushida, Takayuki Okatani, Koichiro Deguchi
Graduate Scholl of Information Sciences, Tohoku University
Aramaki-aza Aoba 6-6-01, Aoba-ku, Sendai 980-8579, Japan

Abstract

In this paper, we present a method for recognition of human activity as a series of actions from an image sequence. The difficulty with the problem is that there is a chicken-egg dilemma that each action needs to be extracted in advance for its recognition but the precise extraction is only possible after the action is correctly identified. In order to solve this dilemma, we use as many models as actions of our interest, and test each model against a given sequence to find a matched model for each action occurring in the sequence. For each action, a model is designed so as to represent any activity containing the action. The hierarchical hidden Markov model (HHMM) is employed to represent the models, in which each model is composed of a submodel of the target action and submodels which can represent any action, and they are connected appropriately. Several experimental results are shown.

1 Introduction

Recognition of human activity in an image sequence is a challenging field of study. It has many applications such as visual surveillance systems for security. In some of the applications, it is reasonable or necessary to incorporate a hierarchical structure in representation of activities, in which a human activity is represented by a combination of multiple activity units, or actions. To be specific, in the rest of the paper, we will use the term *activity* for representing a sequence of *actions*, and an action for representing a component motion, such as "walking", "sitting", "going into a room", "reading a book" and so on.

This hierarchical structure of human activity poses a difficult problem. The data given for a recognition system is merely an image sequence. Thus, in order to recognize each action appearing in the given sequence, it is first necessary to identify the portion of the sequence supposedly corresponding to an action. However, in order to identify and extract the exact portion where the action occurs, it is necessary to understand what action occurs in the portion of interest. This poses a chicken-egg dilemma. Moreover, in a series of actions, the boundary of each action is in general obscure, and thus is difficult to identify; one action can follow the next action immediately, or even the transition is smooth and indivisible.

There are many researches about human activity recognition. Most of them deal with recognition of a single action in the above sense, where the input sequence contains only one action. A few researches dealing with the above difficulty is the one by Ali and Aggarwal [1], in which an activity sequence is segmented into simple actions, such as "walking" and "sitting," by a boundary search of the actions based on a simple visual clue such as the angle of the inclination of the target's body and legs. Obviously, it is difficult to apply to more complex actions. There is also another way of dealing with the problem, which is recognizing an activity sequence directly as a single action pattern. However, depending on applications, it is not practical to make up all possible action patterns because of a huge amount of possible combinations.

In this paper, we present a method that resolves this difficulty. The basic idea is to prepare one model for each action, such that the model can represent the *total* sequence *containing* at least one occurrence of the target action; in other portions of the sequence, any action or any combination of actions may occur. More specifically, each model is designed so that it represents a series of three sub-sequences, the pre-action, action and post-action sequences, and the pre-action and post-action sequences can represent any action or any combination of actions. In a recognition process, a given sequence is tested against every action model in an online manner, and a matched model is searched. When a matched model is found, its corresponding action is identified; not only occurrence of the action but its approximate starting time and duration are also obtained. This search is carried out for every action model, and therefore if there are multiple actions in the given sequence, the method can identify each of them correctly in occurrence order.

In order to implement the above model of three sub-sequences, pre-action, action, and post-action sequences, we use the hierarchical hidden Markov model (HHMM). The occurrence of each action is identified by likelihood of these models. The starting time and duration of the action is measured by the time series transition of the model likelihood.

This paper is organized as follows. A brief introduction of the HHMM is shown in Section 2. The method for identification of actions is given in Section 3. Section 4 shows several experimental results. Section 5 concludes the research.

2 The Hierarchical Hidden Markov Model

The hierarchical hidden Markov model (HHMM) was first proposed in [3] as a hierarchical version of the hidden Markov model. The HHMM consists of three types of states: internal states, production states and

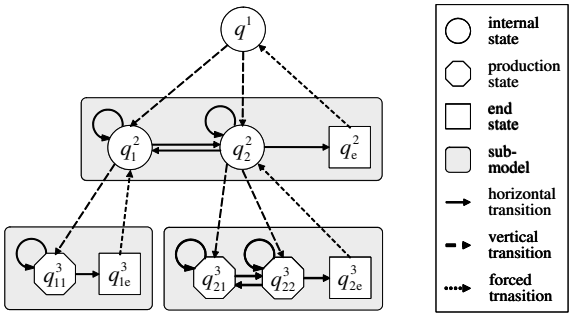


Figure 1: An example of the HHMM with three layers.

end states, and among them there are three types of state transitions: horizontal, vertical, and forced transitions.

Figure 1 shows an example of the HHMM. A state within the HHMM is denoted by q_i^d, where d is the hierarchy level and i is the state index. An internal state may have submodels in its lower layer. The submodel is an HHMM by itself and consists of one or more internal states, or one or more production states and one end state. The production states output symbols in the same way as the states of an ordinary HMM. The end state is the only state where a forced transition arises. The horizontal transition is the intra-submodel state transition, and it is defined only between the states belonging to the same submodel. The vertical transition is an inter-submodel transition and is always downward; it goes to a state in a model of a lower layer. The forced transition is another inter-submodel transition and always gives a path returning to the parent state of the submodel. The transition probabilities for the horizontal and the vertical transitions are defined for each internal state q^d as a vector Π^{q^d} and a matrix A^{q^d}, respectively. The probabilities that a production state outputs symbols are represented as a matrix B^{q^d}.

An HHMM is characterized with its structure and the probabilities Π^{q^d}, A^{q^d} and B^{q^d}. The structure embodies the hierarchy of submodels, the number of states in each submodel, and the possible output symbols. When creating an HMMM, these probabilities, Π^{q^d}, A^{q^d} and B^{q^d}, are determined by learning with given training sequences. The Baum-Welch algorithm [4] is used for this learning, in which the likelihood of the training sequences is maximized.

3 Recognition of Activities

In this section we describe the method to extract a particular action from a given activity sequence by identifying the actions.

3.1 Construction of models of actions

The HHMM framework is used to implement the hierarchical structure of our problem of recognizing actions. In our model of an activity, there are two component submodels. The one is called an action model, which is created per one action to be recognized and represents the specific action. The other is called an universal model, which acts as a wild card and can represents any action or even any combination of different actions. The top-level model, called an activity model, has an action model and a universal model as its submodels in a lower layer. The hierarchical structure is implemented directly by the HHMM.

We represent each action by ω_n, where $n \in \{1, \cdots, N\}$ is the index of actions. For each action ω_n, there is a single corresponding action model, which we denote by λ_n^{a}. Since there are N actions, there are N action models. The structure of an action model λ_n^{a} is manually designed considering the nature of the action ω_n. The probabilities in the designed structure of the action model are determined by learning using training sequences based on the generalized Baum-Welch algorithm.

We represent the universal model by λ^{u}. There is only one universal model in the system. It is also manually designed so that it satisfies the following criteria. First, every symbol is generated with the same probability. In the universal model, the likelihood of occurrence of any action is constant assuming if the durations of actions are the same. Second, the likelihood of occurrence of a particular action ω_n in the universal model is always lower than the likelihood of the same action in its corresponding action model λ_n^{a}, and is always higher than the likelihood of the same action in other action models. Universal model is used not only as a submodel of a top-level model, but as a top-level model by itself, as will be explained in what follows.

As described, an activity model λ_n is the top-level model and is composed of an action model λ_n^{a} and the universal model λ^{u}. The connection of these submodels in an activity model is shown in Fig.2. There are as many activity models as action models, and thus there are N activity models. Each activity model has a single target action, which is characterized by the action model contained.

Because of the redundancy realized by the presence of the universal model, an activity model acquires the following flexibility. When a given sequence is imaginarily divided into three subsequences, pre-action, action, and post-action sequences, the universal model in the activity model deals with the pre-action and post-action sequences, and the action model deals with the action sequence. Thus, the model can represent any sequence such that the action occurs anywhere in it. If a given sequence contains a particular action somewhere in the sequence, the activity model corresponding to the action returns a high likelihood value, from which it is identified that the action occurs in the sequence.

3.2 Recognizing occurrence of an action in a sequence

Using the hierarchical model described above, given an observation sequence $O = \{o_1, \cdots, o_T\}$, we identify actions by calculating likelihood of the sequence for each model. Specifically, we calculate the likelihood of the sequence for each activity model $P(O|\lambda_n)$ and compare it with that for the universal model $P(O|\lambda^{\mathrm{u}})$.

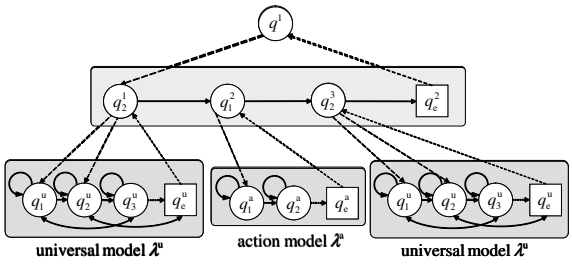


Figure 2: Activity model. q^{a} and q^{u} are states of the action model and the universal model, respectively.

To be specific, we calculate the ratio

$$P(O|\lambda_n)/P(O|\lambda^{\mathrm{u}}) \tag{1}$$

at each time step of the sequence and check if the ratio exceeds a threshold P_{th}. If the ratio exceeds the threshold, we identify the action occurs there. The time step at which the ratio exceeds the threshold gives an approximate time of the occurrence of the action (more likely, the end of the action).

3.3 Extracting the subsequence associated with the recognized action

In the above method of identifying actions in a given sequence, approximate time of occurrence is obtained in addition to their occurrence. However, they might not have sufficient accuracy depending on applications in which more accurate start and end of the action are necessary. We present a method for extracting the accurate subsequence of the action.

Suppose that we have already identified the occurrence of a particular action in a given sequence. Then, the start and the end of the action are searched using the probability of the action occurrence.

Suppose a current state in the activity model corresponding to the time step t. There is a single state corresponding to each time step (or the symbol at each time step), which is determined only in a probabilistic sense. Then, in order to define the start of the action, we calculate the probability that the current state is one of the states belonging to the action model of the activity model, as follows:

$$P_s(O|\lambda_n) = \sum_{q \in \lambda_n^a} p(\text{the current state is } q|o_1, \ldots, o_n). \tag{2}$$

This can be evaluated using the probability structure of the same model used in the method in the last subsection. Then, we define the start of the action by the time step at which the above probability exceeds a threshold determined in advance. Also in order to identify the end of the action, we calculate the probability that the current state is the end state of the associated action model, as:

$$P_e(O|\lambda_n) = p(\text{the current state is } q_e|o_1, \ldots, o_n). \tag{3}$$

Then we define the end of the action by the time step at which this probability attains its maximum value.

Because of the parallel structure of the assumed models in the above method, it sometimes happens that two or more different action models wrongly recognize two or more different actions simultaneously, that is, those different actions are regarded to occur at the same time. In that case, it necessary, it is possible to select the most likely single model, by selecting the model that yields the maximum possibility.

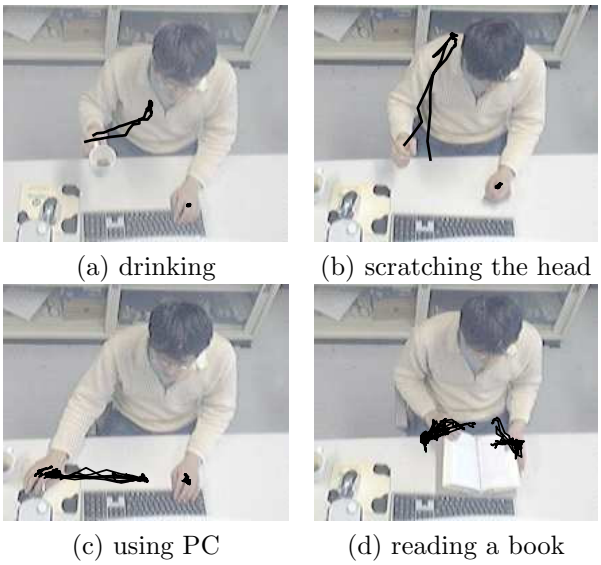
4 Experimental Results

We applied the method to the problem of recognizing activities of a person sitting in a chair in front of a desk. The recognition is performed based on the subject's hand motion on the desk surface. Specifically, trajectories of the subject's hand are observed in an image sequence and used. In order to obtain the trajectories of hand motion, we use a tracking algorithm based on the mean shift algorithm that tracks a region of skin color in the image [2].

The trajectories extracted are transformed into a sequence of symbols. The image plane is divided into 8×6 rectangular regions, and symbols are assigned to each of them. Time is quantized using a constant interval. Then, using the position of the subject's hand at each time step, the sequence of the associated symbols is obtained from the trajectory. When the same symbol appears iteratively for many times for a given trajectory, they are suppressed into a shorter iteration of the symbol. Thus, the index of a symbol in a sequence and its time step at which the symbol appears correspond with each other in nonuniform manner. By this method, computational complexity is considerably reduced. Precisely, when there is a symbol repeatedly appearing for k successive times, the k succession of the symbol is represented (replaced) by a $\log_2(k+1)$ succession of the symbols.

We define four actions and use them for recognition: "reading a book", "using PC", "drinking", "scratching the head. " A sample image and trajectories of some of these actions are shown in Fig.3. Activity models of these actions are created as described in Section 3.1, where N training sequences are prepared and used. The training sequences are such that each sequence has only a single target action. The thresholds are manually chosen based on the results of the training sequences.

Then a set of test sequences is acquired and used for recognition experiments. Figure 4 shows an example of a recognition result for a given sequence. The figure shows time-series variation of the log-likelihood ratio (1) about P_s and P_e whith respect to the sequence for each of the four activity models. As is shown, the likelihood ratio for each activity model varies at each symbol. In the experiment we set the threshold to be 10.0; occurrence of the action is identified by checking whether the ratio of P_e has a maximum value larger than 10.0. Once an action is identified by this thresholding, the start of the action is recognized by searching the previous time step at which the ratio of P_s exceeded 0. The intervals shown in the top of the plots by the double-edged arrows show the actions ac-



(a) drinking (b) scratching the head

(c) using PC (d) reading a book

Figure 3: Example of the trajectories for the actions to be recognized.

tually performed by the subject. Those shown in the bottom of the plots show the actions recognized by the proposed method. It can be seen that except for a few portions in which the recognized actions overlap with each other, the recognition results mostly coincides with the true actions performed. The method is applied to many sequences and the results are checked. Table 1 shows the rate of successful recognition.

Table 1: Recognition rate.

action	drink	scratch	PC	read
rate	1.00	0.88	0.92	0.90

5 Summary

We have shown a method for recognizing human activities from a given image sequence. The motivation of this research is to deal with the difficulty with recognition of a sequence in which multiple actions can occur in any possible order. In order to resolve the difficulty, we create as many models as actions to be recognized, and test each model against the given sequence to find a matched model. The label of the matched model yields the action performed in the sequence. Since this test is done in a parallel manner for all the models, the method works well when any combination of actions occurs in any order in the sequence. The search of the matched model is done by using the likelihood of the sequence for each model. For each of the actions to be recognized, we construct a model called the activity model, in which there are three submodels, a model of the target action, and two models for representing any action called the universal model. Thus the model has a hierarchical structure, and the HHMM is used for implementing this structure. We have shown preliminary experimental results, from which the proposed method

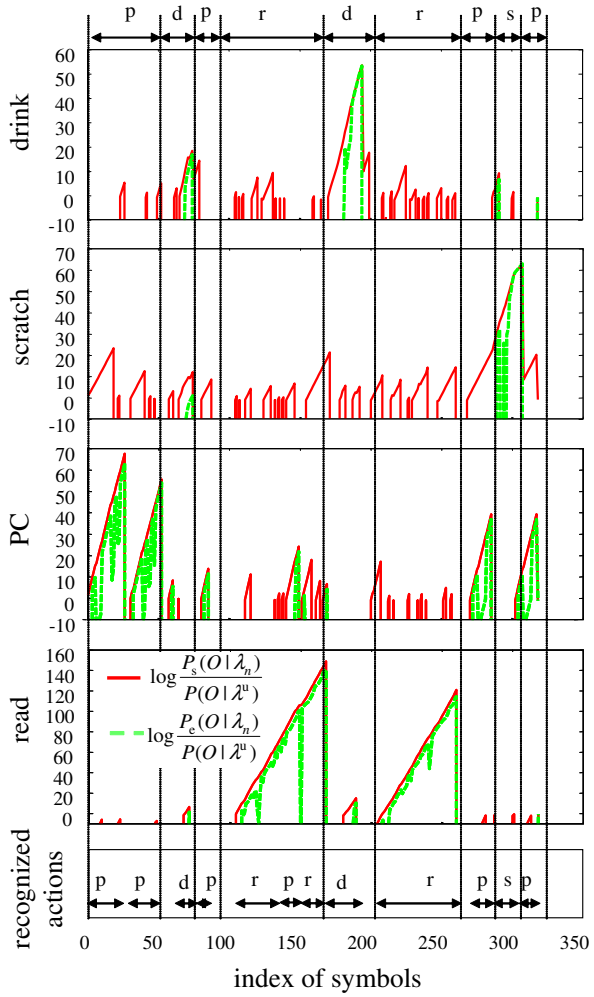


Figure 4: Likelihood (precisely a likelihood ratio) of a given sequence for each of the activity models. The intervals in the top show the actions performed by the subject, and those in the bottom show the actions recognized using the likelihood: p: using PC, d: drinking, r: reading a book, and s: scratching.

does work well for the sequence in which multiple actions occur.

References

[1] A. Ali, J. K. Aggarwal, "Segmentation and Recognition of Continuous Human Activity" *In IEEE Workshop on Detection and Recognition of Events in Video*, pp.28-35, July, 2001

[2] D. Comaniciu and P. Meer, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, no.5, pp.564-575, 2003

[3] S. Fine, Y. Singer and N. Tishby, "The hierarchical hidden Markov model: Analysis and applications", *Machine Learning*, vol.32, pp.41-62, 1998

[4] S. Lühr, H. H. Bui, S. Venkatesh, G. A. W. West, "Recognition of Human Activity through Hierarchical Stochastic Learning", *PerCom'03*, pp.416-422, March 2003