Learning Human Behaviour Patterns in Work Environments

Chih-Wei Chen  
Stanford University  
Stanford, USA  
louistw@stanford.edu

Asier Aztiria  
University of Mondragon  
Mondragon, Spain  
aaztiria@eps.mondragon.edu

Hamid Aghajan  
Stanford University  
Stanford, USA  
aghajan@stanford.edu

Abstract

In this paper, we propose a flexible, human-oriented framework for learning the behaviour pattern of the users in work environments from visual sensors. The knowledge of human behaviour pattern enables the ambient environment to communicate with the user in a seamless way and make anticipatory decisions, from the automation of appliances and personal schedule reminder to the detection of unhealthy habits. Our learning method is general and learns from a set of activity sequences, where the granularity of activities can vary for different applications. Algorithms to extract the activity information from the videos are described. We evaluate our method on video sequences captured in a real office, where the user’s daily routine is recorded over a month. The results show that our approach is capable of not only identifying the frequent behaviour of the user, but also the time relations and conditions of the activities.

1. Introduction

Intelligent Environments (IEs) are expected to support people in their daily lives. A basic assumptions about IEs is the change of perspective in the relationships between humans and technology, shifting from a techno-centered perspective to a human-centered one. This implies that IEs must support people in their daily tasks, providing personalized services adapted to their needs. Thus, IEs can be defined as “digital environments that proactively, but sensibly, support people in their daily lives” [1]. Let us consider the following scenario:

Michael arrives at his office around 9 a.m., and goes directly to his desktop. By then, his computer is automatically switched on. After checking his email, he meets with his coordinator to plan the work to do. Then, he goes back to his desk and sometimes he prints his schedule. As the environment has already learned his common behaviour, it is able to send him reminders if it detects that Michael may have forgotten an activity. Besides, the environment can suggest him to take a break if he is working longer hours than usual.

Figure 1: Action Map of Michael’s morning ritual.

Human-centered computing has seen great progress in recent years [7], but the adaptation of the environment should go beyond the traditional multimodal human-computer interaction framework. The environment should act proactively in order to adapt itself to the users and provide personalized assistance. For that shift to take place, the environment must know users’ habits and needs, discovering such patterns in an unobtrusive and transparent way.

Assuming that human beings perform behaviours based on habits, it could be inferred that patterns describing past and present behaviours will define future behaviours as well. In that sense, discovered patterns can be used to proactively activate/deactive some devices (e.g., switching on the computer automatically when Michael goes into the office). Apart from automating actions or devices, patterns can also be used to understand his behaviour and act in accordance with it (e.g., issuing reminders). Therefore, unhealthy habits can be detected (e.g., his work periods are too long). Making the environment more efficient in terms of saving energy (e.g., switching off the lights when he has gone out) or increasing safety (e.g., locking the door when he has gone out) are other aspects that can be supported by an IE when it is aware of users’ frequent behaviours.

While sensors of multiple modalities can be used to discover behaviour patterns, in this work we focus on visual sensors because of their ability to observe the users in an unobtrusive way. We develop a system consisting of a visual processing and a learning module that addresses the following challenges: extracting information from the visual sensors reliably and discovering the pattern efficiently. The underlying visual processing module is capable of tracking multiple people in a realistic environment. The trajectory of the user allows location-driven routines to be identified.
We simplify the behaviour discovery problem by associating the semantic locations of the user to activities. While more detailed activity categories can be discovered by human activity recognition techniques, this association is effective in a workplace scenario where activities are strongly correlated with locations by their functionalities. For activities taking place in close proximity, we use a clustering algorithm on the visual data to separate activity classes into finer granularity. The locations and the clustering results are mapped to semantic activities. Our Learning Frequent Patterns of User Behaviour System (LFPUBS) then learns the user’s common behaviours. The objective of LFPUBS is to discover accurate patterns that represent the user’s frequent behaviours in a comprehensible way by means of Action Maps. This model allows the expression of time relations using relative time references. Following our previous scenario, Michael’s morning behaviour represented in an Action Map is depicted in Figure 1.

Our work investigates the utilization of visual sensors in learning comprehensive behaviour patterns. The contributions of this paper are as follows. First, the proposed visual processing algorithms extract locations and activity modes that are characteristic of people’s routines. Second, we apply a data mining approach for the discovery of the behaviour patterns, including time relations and conditions. Finally, we show that more detailed behaviour understanding can be achieved if our system learns from activities of finer granularity. The proposed framework is general, not restricted to work environments and can be extended to learn from a richer set of activities.

2. Related Work

Understanding human behaviour has attracted a significant number of researchers, and much work has been devoted to modeling human behaviour at different temporal resolutions, from pose estimation and activity recognition to long term behaviour pattern learning. In particular, spatial-temporal based [9, 14] or shape based [12] human action recognition algorithms have been developed for the understanding of human motion and activity. Multimodal sensors in ambient intelligence environments [2, 5] or in mobile phones [6] have been integrated to discover patterns in diverse spatial and temporal scales, from a specific activity such as cooking to a person’s daily life over a timespan of months. Other groups have tried to recognize affective and social signals in order to create anticipatory interfaces. A survey can be found in [10]. Our work learns comprehensive behaviour patterns, and can incorporate the activity recognition or multimodal results to enrich the model.

Contextual information has also been extensively used in the recognition of human poses [8] and pattern discovery [3]. Understanding the pattern of human activities is also beneficial for activities recognition. It has been shown that prior knowledge about the structure of human activities provides constraints in the learning process and better accuracy can be achieved [16].

3. User Location and Activity

Given the daily videos recorded in the work environment, our system first extracts the locations of the users. The information is subsequently sent to the learning layer for the discovery of behaviour patterns. Here we describe the vision tracking algorithm and the motion clustering method used to label the users’ activities.

**Tracking** For tracking the user in the environment, we use a tracking-by-detection approach similar to that used in [4]. The design philosophy is to build an efficient yet robust tracker that runs in real-time to meet the interactive nature of the overall system. Our tracking framework integrates two complementary methods: detection association and low-level tracking. For detection, we construct an adaptive background mixture model for the incoming video, and obtain a binary foreground mask for each frame. The foreground and background segmentation not only reduces the search space for detection, but also provides an extra constraint: only detection near the top of a foreground blob is valid, since the target is the head of a person. The Ω-shape head and shoulder silhouette template is then matched against the edge map of the frame with the foreground mask applied. The template matching is performed at multiple scales using Chamfer distance, therefore robust to cluttered backgrounds and appearance variations.

Formally, let $T$ denote the template, $I$ the image frame, and $E$ the edge map. The Chamfer distance at location $x$ is calculated as,

$$d(x, T) = \frac{1}{|T|} \sum_{x_t \in T} \min_{x_e \in E} \| (x + x_t) - x_e \|$$

where $|T|$ is the number of points in the template $T$ and $\| \cdot \|$ can be either $l_1$ or $l_2$ norm. The Chamfer distance is efficiently computed by first computing the distance transform on the edge map $E$.

Now let $x_t$ and $s_t$ be the location and scale of the tracked target at time $t$, and $b_t = (x_t, s_t)$ denotes the bounding box around the target. We update the track as follows,

$$b_{t+1} = \min_{b_d \in D} d(x_{d}, T_{s_t}) + \alpha \cdot corr(b_t, b_d) + \beta \frac{b_t \cap b_d}{b_t \cup b_d}$$

where $D$ is the set of detections returned by the head detector at time $t + 1$, $T_s$ the template at scale $s$, $corr(b_1, b_2)$ the normalized correlation of the image patch inside $b_1$ on $I_t$ and that inside $b_2$ in $I_{t+1}$. The last term calculates the overlap between the candidate detection and the previous bounding box, and $\alpha, \beta$ are weighting parameters.

48
When there is no detection, possibly due to no motion (foreground person merges into background model) or heavy occlusion, our system falls back to low-level tracking. We search in a local region and find the image patch in the new frame that achieves maximum normalized correlation with the current target image patch. If the best match has a matching score below a predefined threshold, we do not update the location and keep the previous image patch as the appearance template. This allows the track to be associated with new detections when they reappear.

Once the head location is found, the 2D coordinate of the user within the environment can be estimated given the height of the user and the camera calibration parameters. The 2D location is then quantized into semantically meaningful regions, where each region is defined manually and is weakly associated with different user activity by its functionality. Examples of the region labels include personal desks, printers, and coffee machines, which correspond to work, printing documents, and making coffee. Finally, the representation of the user’s location is the region label.

Motion Clustering While many activities are highly correlated to or even constrained by locations, there are situations that cannot be disambiguated from the sole perspective of location. For example, when the user is at his desk, he might be working on the computer or on papers. These distinct activities have various implications and represent different state in the behaviour pattern. To further understand a person’s routine, we propose to incorporate additional visual sensors in the environment to provide richer information.

Our approach to distinguish different activities is inspired by the extensive research in human activities recognition. In particular, we focus on activities over a short temporal window, represented by a set a measurements [15]. The measurements we consider are motion and appearance, represented by motion history image (MHI) and color-based skin detection. The latter has been widely used for the detection of human, as well as describing the shape and suggesting the pose of the human body.

To classify an observation, representative exemplars are sampled for each class of activity from a training video. Each exemplar is represented by the motion history image and skin detector result. To accelerate the matching process, principal component analysis (PCA) is employed to reduce the dimensionality of the data. The distances from the new observation to the exemplars are then calculated in the reduced feature space. The final assignment is determined by its $k$-nearest neighbors ($k$-NN).

4. Learning User’s Frequent Behaviours

Once the user’s activities are labeled by the visual processing module, the learning module discovers the hidden frequent behaviours. LFPUBS consists of two submodules. The underlying idea is the separation of the representation of the discovered patterns from the process of discovering per se. The core of the representation module is a language ($L_{LFPUBS}$) that provides a standard conceptualization of the patterns so that the environment is able to represent all type of patterns in the environment. On the other hand, the process of discovering is based on an algorithm ($A_{LFPUBS}$) that discovers frequent patterns.

4.1. Representing patterns with $L_{LFPUBS}$

Because of the complexity of IEs, defining a language that allows the environment to represent discovered patterns in a clear and unambiguous way is difficult but necessary. The language integrated within LFPUBS is based on ECA (Event-Condition-Action) rules. A frequent behaviour is defined by means of an Action Map, which contains all of the specific relations between actions. In other words, an Action Map is created relating actions in pairs, and each of those relations is called an Action Pattern, which are defined by means of ECA rules. The complete Action Map is then created by linking different Action Patterns.

In the same way as ECA rules, each Action Pattern relates two actions (defined by the ON and THEN clauses) and the specific conditions (defined by the IF clause) under which that relation occurs. Finally, unlike basic ECA rules, $L_{LFPUBS}$ allows the environment to define the time relation between both actions. Considering Michael’s behaviour of having meetings. Using $L_{LFPUBS}$ it would be represented as follows:

(Action Pattern 14)
ON occurs (simple,(Working,off),t0)
IF context ()
THEN do (simple,(On,Meeting),t) when t = t0 + 3s.

4.2. Learning patterns with $A_{LFPUBS}$

Coupled with $L_{LFPUBS}$, an essential component in the Learning Layer is the algorithm ($A_{LFPUBS}$) that discovers the frequent behaviours of the users. The different steps to be performed by $A_{LFPUBS}$ in order to discover the frequent behaviours of the users are depicted in Figure 2. A brief description of each module follows:

Figure 2: The system diagram of the learning algorithm.

Identifying Frequent Sets of Actions The objective of this step is to discover the sets of actions that frequently occur together (Frequent Sets). The underlying idea of the first step is both simple and efficient. Defining a demanded minimum level (minimum confidence level), it discovers all those sets of actions that occur more times than
Identifying Topology The step Identifying Frequent Sets discovers which sets of actions frequently occur together. In order to properly model the user’s behaviours defined by such sets of actions, it is necessary to define the order of such actions. That is the goal of this step, to discover the frequent order (defined as Topology) of the actions in the behaviour of the user. For that, we use techniques applied in the Workflow Mining area [11], where models are discovered from event logs. Even so, because of the nature of IEs, some particularities must be taken into account.

Unlike other domains where actions are unique, in IEs there could be different occurrences of the same action. In fact, the nature of repetitive occurrences will probably be different because the user can do the same action with different purposes. The technique to discover repetitive actions is based on the idea that the meaning of an action is mainly defined by the previous and next actions. Thus, we collect the previous and next actions, and using the EM algorithm we define the number of occurrences we need for each action. In Michael’s case, the only actions that need more than one occurrence are the ‘Working On’ and ‘Working Off’ actions.

Identifying Time Relations Although the topology defines sequential relationships between actions (defined by the qualitative term ‘after’), more accurate time relations are necessary. The objective of this step is to discover frequent quantitative Time Relations between the actions defined by the topology.

Qualitative relations allow one to understand the logical order of the actions. Even so, quantitative Time Relations provide higher quality information. For example, we might want to automate a device such as switching on Michael’s computer. If such a relation is defined by means of a qualitative Time Relation, the environment would not know when it has to switch on the computer without the knowledge of whether the time delay was 2 seconds, 5 minutes or 2 hours. However, using quantitative relations (3 seconds in Michael’s case) allows the system to switch on the computer of the meeting area at the right time.

Quantitative Time Relations are discovered using clustering techniques. If any cluster covers more instances than a demanded minimum level, the mean value of the cluster is considered as quantitative Time Relation. In Michael’s case, this step discovers that he usually needs 3 seconds to reach the meeting area when he leaves the working area. It also discovers the duration of each of the work periods, giving a quantitative value to those periods.

Identifying Conditions Finally, in order to achieve complete patterns, it is necessary to discover under what conditions each relationship occurs. It could happen that an action is followed by two (or more) different actions, so that it is necessary to define under what conditions the user carries out each one of the actions. In Michael’s case, this step discovers when he prints his schedule and when he does not.

In order to identify conditions, for each possible relationship a table is created. In each table the occurrences covered by that relationship are collected, together with the calendar and contextual information collected when such occurrences happened. Once the tables are created, separating tables by using the information they contain allows one to discover conditions. In that sense, the task of separating can be solved by treating it as a classification problem. The JRip Algorithm [13] was used to accomplish this task. It discovers that Michael prints his schedule if he has meetings before 9:15 a.m..

5. Experiments

The experiments to validate the visual processing module, as well as the learning system, are carried out in a real office environment. During a period of a month, the system recorded 19 days of morning behaviour of an employee who behaved in his usual manner, i.e. without any pre-defined behaviour or modifying his routine. Data were collected using Axis network cameras, capturing VGA (640 × 480 pixels) videos at 30 fps. Two cameras were used in this experiment: one camera installed on the wall for tracking the location of the user, and the other dedicated to monitoring the desktop. Our tracking algorithms process the videos at 6 fps on a laptop with a 1.66 GHz Core Duo processor.

Visual Processing We first evaluate the performance of the visual processing module. Groundtruth for one 35 minutes long sample video from the recordings is manually labeled. A detection matches to a groundtruth if the overlapping area is at least 50% of both rectangles. On the test sequence our tracker achieves 98.3% accuracy. The track is lost mainly under two situations: if the target person moves too fast and not in the upright position (thus no detection), or if the person moves his chair while his head remains static. In the latter case, the detector might be confused by the moving chair’s back, which resembles the silhouette of a person’s head and shoulder. Examples of the tracking results obtained from all videos are shown in Figure 3.

The 2D location is calculated from the head location in the video frame. We assume that the person is either standing or seated, and the person is standing during the initialization of the tracking. In most cases, only one of the height assumption would lead to valid location within the office. In case of ambiguity, the location closer to the person’s previ-
Figure 3: (a) Office layout The office is quantized into semantic areas by their functionalities, including Working, Meeting, Printing, Door and Special. (b) Tracking results The composite of tracking results from several days of the user in different areas. The bold red bounding boxes show true positive on the target’s head, while the thin green bounding box indicates false positive. In this case the tracker mistakes the chair for a person’s head and shoulder, which shares similar silhouette. (c) Desktop camera The user is in the same location but doing different activities. The extra visual sensor helps disambiguate different activities that cannot be classified by location alone.

Figure 4: Localization results for a half-hour long sample video. The groundtruth location of the target over time is shown. The boldface number in parentheses indicates the error (in seconds) of our localization algorithm. No number preceding the time label means correct results. Our algorithm is capable of identifying the location of the user with low temporal error.

ous position is selected. The result from the same test sequence used above is shown in Figure 4. As shown in the figure, the locations of the user are correctly identified by our proposed method. When the tracker fails to follow the target immediately after he leaves a region, the localization cannot follow until the tracker recovers. Another source of error comes from projecting the image point back to the real world coordinate. For examples, the printer is located at the far end of the room from the camera. Even small error in head location estimation would be amplified and leads to spurious 2D localization.

Finally, we evaluate the motion clustering on the second camera. An extra sequence not in the test set was recorded to collect the activity exemplars. In this project we consider two activities at the desk: working on the computer vs. on papers. Example frames are shown in Figure 3. Using k−NN, we achieve 82.7% accuracy on 7436 frames of the user at the desk in the test sequence.

Behaviour Learning Once user’s actions were identified, we run LFPUBS in order to discover his frequent behaviours. In fact, the example we have used throughout the paper (see Figure 1) is the behaviour of the experiment subject discovered by LFPUBS. Each step, together with the runtime, of the learning process is analyzed in more detail:

Identifying Frequent Sets of Actions (26 ms.) The demanded minimum confidence level is set to 50%, which means that a behaviour has to occur at least in 50% of the days to be considered as frequent. This step discovered one Frequent Set of Actions that contains the actions ‘Working On’, ‘Working Off’, ‘Meeting On’, ‘Meeting Off’, ‘Printing On’ and ‘Printing Off’.

Identifying Topology (215 ms.) This step is essential to providing comprehensible pattern because the employee usually carried out the same action (e.g. ‘Working On’) sev-
eral times with different purposes. In that sense, the process of identifying repetitive actions was critical to create meaningful patterns. LFPUBS was able to infer the need of 3 different occurrences for the actions ‘Working On’ and ‘Working Off’.

Identifying Time Relations (210 ms.) The demanded minimum level to consider a Time Relation was 25%, and LFPUBS is able to identify quantitative Time Relations in 6 out of 9 relationships. This is the critical step to identify, for example, how long the employee worked without any break, which subsequently allows an expert to decide if he needs to adjust his behaviour. In our user’s case, LFPUBS discovered that he usually worked for 25 minutes without any break after he printed his schedule.

Identifying Conditions (68 ms.) Finally, LFPUBS was able to identify conditions on which the user printed his schedule or not. Due to the lack of other types of information (e.g. agenda, contextual, etc.), the only available information was related to time. In our case, the user printed his schedule if he had had meeting before 9:15 a.m.

Incorporating Additional Visual Sensors Besides identifying the frequent behaviours, the incorporation of additional visual sensors allows us to analyze the user’s behaviour with finer granularity. Temporal smoothing is applied to the motion clustering results. Based on the additional information, LFPUBS is able to analyze the working period in more details and discover that while the user constantly stays in front of the computer, he sometimes moves away from the screen and works on papers (13 minutes on average).

6. Conclusion

Intelligent Environments suggest a new paradigm where environments adapt their behaviour based on needs and habits of users instead of the other way around. For that, environments should learn, without disturbing the users, their behaviour patterns. To achieve that goal, we have developed a system that is able to discover user’s frequent behaviours. Our proposed framework has proved its ability to discover accurate and comprehensible patterns. The patterns are represented by means of Action Maps that allow one to define the order of the actions as well as the time relations between such actions and the conditions under which such relationships occur.

Although the categories of human activities considered in our experiments are limited and require manual assignment, our proposed framework is general and flexible. More complex patterns can be learned by incorporating multimodal sensors and different activity recognition techniques. Other directions of future work include: learning behaviour and interaction pattern of multiple people; detection of behaviour deviation from learned patterns; using a richer set of contextual information.

Acknowledgments This research was partly funded by the Department of Energy under grant DE-AR0000018.

References