國立臺灣大學電機資訊學院資訊工程研究所

碩士論文

Graduate Institute of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

以情境地圖達成在辦公室環境中以人為中心的服務

Human-Context Mapping for Human-Centric Robot Service in Office Environment

俞冠廷

Kuan-Ting Yu

指導教授：傅立成 博士

Advisor: Li-Chen Fu, Ph.D.

中華民國 101 年 7 月

July, 2012

誌謝

令人緊張的口試結束，首先感謝傅立成老師在研究過程中不斷指引方向，讓我勇於探索一些重要且未知的領域，並且讓研究成果在最後能夠收斂得更加完美。感謝評審委員王傑智老師、簡忠漢老師、李蔡彥老師、黃正民老師用心給予的建議。王老師在我學習和研究過程中也很熱心地幫助我，精闢地講出我可以加強的部分。徐宏民老師在我碩二旁聽課程時，特別照顧我，在討論問題時讓我感受到研究的樂趣。

感謝碩零的仲達、小朱、亦修，在最後關鍵的幾週不斷幫我做一些「奇怪」的實驗，你們積極的態度，讓我十分感動。夏天午後，在冷氣房外跟你們一起流汗做實驗的時候真的很開心。感謝佩文在機器人手臂上的幫忙，才能完成最後的倒咖啡服務。感謝元翰在機器人導航的幫忙，讓她-艾薇爾-能夠順利得在實驗中穩定行走。感謝符劼花了許多時間協助精進我的人員偵測程式，讓我可以進行多人的實驗。感謝士桓大學長在研究上的討論，還有讓投影片更加精進。感謝峯志在語音功能上的幫忙，讓艾薇爾能夠在實驗中可以流暢地聽說。感謝三位碩零新生、明芳、佩文、聖諺、建科、元翰當作訓練資料的模特兒。感謝碩二一起奮戰的戰友們，雨喬、船姊、稟哲、偉豪、依書、大頭，讓我覺得碩二不是很孤單。最後感謝機器人艾薇爾在這段期間不斷被測試到手腕斷掉也無怨無悔。

感謝我的父母親，在研究所期間幫我準備午餐便當，讓我可以無憂無慮吃得健康。感謝沂霏在研究時到實驗室陪伴我，讓我安心，並且在機器人之外有人可以講話。

另外，在孤獨的研究生活中，我要感謝三位歌手讓我繼續充滿動力。Lady Gaga 教我在研究和生活中走出自己的路，廣仲讓我學習用正面態度面對困難，杰倫教我在挫折時仍懂得珍惜，知足最快樂。

冠廷

2012/7/27

中文摘要

服務型機器人在服務使用者時，必須能夠考慮到人的情境狀態，才能自主地提供符合情境的服務。在這篇論文中，我們專注於辦公室的使用者情境，因此定義了六種使用者狀態：專注、疲累、放鬆、休息、社交以及其他。為了要讓機器人從影像觀察中推論使用者的情境，我們提出了一些創新且具鑑別力的特徵。這些特徵包含使用者姿態、人與物體的互動和人與人的互動，而後面兩者為目前較少被用於使用者狀態辨識。

此外，為了更有效地推斷使用者情境，我們提出將使用者情境融入地圖建置。所以使用一個空間-時間的格網地圖來記錄使用者情境與地點和時間的關係。我們使用動態貝氏網路 (Dynamic Bayesian Network) 來作為建置地圖與推論的基礎架構。使用情境地圖架構來推測使用者情境，總共有三項優點 1) 機器人可以動態地根據該區域的使用者情境來調整自己的行為，2) 因為在辦公室環境中，使用者通常有固定的行程，所以使用者行為模式可以從先前的情境觀察中累積，3) 當有多個機器人存在於環境中的不同位置時，可以很容易地分享它們的觀察資訊。

另一方面，機器人的行為決策也整合進同一個架構，形成一個動態決策網路，如此機器人可以隨使用者狀況的變動來規劃要提供適當的服務。實驗部分驗證了使用者情境辨識、地圖建置以及整個系統的有效性。

關鍵字：情境地圖建置, 情境感知, 辦公室機器人。

Abstract

Robot that services humans must consider human context in order to behave and service properly. Here, the context categories we tailor for office environment include *concentrating, tired, relaxed, napping, social*, and *neutral*. We design several novel features for inferring human context from visual observation. The features incorporate human pose, human-object interaction, and human-human interaction, of which the last two have rarely been explored in human status estimation.

Moreover, to infer human context more efficiently, we propose a novel semantic mapping framework that embeds human context into a map representation. This produces a spatial-temporal grid map that represents the relationship of human context with location and time. We construct the framework using Dynamic Bayesian Network. There are three exclusive advantages for building such map: 1) a service robot can dynamically adjust its behavior and plan services based on the estimated context in an area; 2) because in office environment people tend to have fixed schedules, people's living patterns can be extracted in the map; and 3) multiple robots can easily share their observations on human context at different places.

On the other hand, robot behavior decisions are integrated into the framework, which leads to a unified Dynamic Decision Network. Thus, the robot can plan proper services according to the human-context map. The effectiveness of our proposed context recognition, mapping and decision framework is verified with simulation and real testing scenario.

Keyword: human context mapping, context-aware, office robot

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Mapping is the most common way to relate events and locations. In robotics field, researchers have been studying Simultaneous Localization and Mapping (SLAM) extensively, whose aim is to autonomously build a metric map or topological map for robot navigation. However, navigation is not enough for robots to perform higher-level tasks in servicing. One promising solution is to incorporate semantic content into the map, which leads to the study of semantic mapping.

Semantic mapping is defined as a branch of robot mapping research which is concerned with the study of building a map embedded with semantic information. In recent years, there has been an increasing interest in this topic to enable robots to perform higher-level tasks that require deep understanding of the environment. The major researches can be divided into three categories. The first group is building a map with small object identification (*e.g.* coffee mugs) or with labels of medium to large structures (*e.g.* walls, doors) [1-10]. The second is labeling functional name of an area (*e.g.* kitchen), so that the robot can perform certain task (*e.g.* go to the kitchen first to fetch a coke from the fridge) [11-14]. The third is adding a conceptualization that is comparable to human understanding of the environment, such as forming "*has-a*" relationship among objects (*e.g.* kitchen *has a* fridge) [13-16]. However, the semantic mapping techniques mentioned above only provide information for robots to plan tasks, manipulate objects or navigate, not for robots to understand human-context and to actively provide appropriate services.

In this thesis, we demonstrate that embedding human-context information into a spatial-temporal map is an efficient way for robots to perform proper behavior for three reasons: 1) service robots can adapt to dynamic context of humans in an area; 2) robots can make predictive planning by inferring human context using previous estimates as prior; and 3) multiple robots can easily share their observation on human context at different places.

Notice that it is the human inside that gives the meaning of the room. Also, human activity is dynamic, so assigning a fixed functional label of a place prohibits robots from adapting to the changing context. For example, when people are concentrating on their discussion in a meeting room, the robot should not disturb them unless it has urgent messages to deliver or is asked for help. On the other hand, as people are having a tea party in the same meeting room, the robot should actively serve drinks and entertain people. This example illustrates that the same place (or places with the same functional label) does not imply the same human context. Therefore, learning from human is a more adaptive way than room recognition for robots to provide context-aware services. One popular solution is applying activity/emotion recognition techniques to understand human context in time [17, 18].

Current methods for activity recognition are mainly based on key poses or actions. However, in practice, discriminative poses or actions may not appear timely for the robot to recognize (*e.g.* wassail for social context). The fact that existing methods typically handles deliberately displayed and exaggerated expression was pointed out in [19]. Moreover, current recognition accuracy is still far from achieving human-level performance. Our work is complementary to researches on activity recognition by exploring the relationship of human context with spatial and temporal domain. The insight here is that people tend to have certain context at certain location and at certain time. This phenomenon is more evident in office environment, where people tend to have fixed schedules. Putting this constraint into consideration, we believe that robots can make predictions on human-context more accurately. As a result, robots can plan their services more efficiently, and the efforts on gathering human-context information can be reduced.

1.2 Related Work

Here we review two topics related to our work, one with semantic mapping, and the other with human context estimation.

1.2.1 Semantic Mapping

Semantic mapping research can be divided into three categories: 1) mapping with object identification or structure labeling; 2) functional name assignment to places; and 3) creating a conceptualization that is comparable to human understanding of the environment.

1.2.1.1 Object-based semantic mapping

In this category, studies can be characterized by the sensor used, consideration of dynamic environment, and improvement on SLAM using semantic information. In terms of sensors, the most frequently used are 3D range sensors, 2D cameras, RGB-D cameras, and gripper.

1) Using 3D laser scanners

Nüchter *et al.* proposed a system that labels large structure such as door, wall, and ceiling with a constraint network [5]. On the other hand, delicate objects are recognized using a trained classifier. In [7], Rusu *et al.* developed a system for creating an object map of a kitchen scene. They proposed a histogram-based feature, FPFH, for conducting registration [20] and object recognition [21] on point cloud data.

2) Using monocular cameras

Civera *et al.* [1] proposed a real-time monocular SLAM system that incorporates object recognition. To recognize objects, the system is loaded with object models described by sparse SURF feature points. Its key advantage is that the system only requires sequence of 2D images as input and operates in real-time.

3) Using RGB-D cameras

The research team from Washington University focuses on using RGB-D camera from mapping to semantic labeling. Here, three recent works are listed as follows. First, in [22], Henry *et al.* made use of RGB-D camera to build dense 3D modeling by combining the visual and shape information. Second, in [8], Lai *et al.* combined 2D view-based recognition approach, 3D Markov Random Field (MRF), and multiple views along time to label small objects in a RGB-D video. Third, Ren *et al.* [10] used kernel descriptors for RGB-D patch feature matching, a superpixel MRF and a segmentation tree to model contextual information for scene labeling.

Nascimento *et al.* [3] proposed an RGB-D descriptor by encoding color and depth information into binary strings, which achieves faster processing time compared to SURF and BRIEF descriptors both in training and testing with an Adaboost classifier.

4) Using grippers

In [2], Blowdow *et al.* designed a semantic mapping system that uses both depth and color information as well as robot's interaction with environment. The captured point-clouds were first registrated using Iterative Closest Point algorithm (ICP). Afterwards, hypotheses of movable parts such as doors and drawers in the map were generated. Then, the robot validated those hypotheses by manipulating those parts using its gripper.

In terms of using semantics to improve SLAM, Rogers *et al.* [4] argued that using complex landmarks such as signs, objects, and furniture in data association have more advantages over low-level features, i.e. ambiguous matching can be reduced. They presented their idea by including a door-sign reader for data association in a graphical SLAM framework.

For dynamic environment, Anguelov *et al.* [6] considered non-stationary objects in an office environment by proposing an EM algorithm for discovering those objects and learning their models. Wang *et al.* [23] proposed a framework of Simultaneous Localization and Mapping with Moving Object Tracking to fully utilize the information from both static and moving objects. Local grid maps are used to record the laser contour of moving objects.

1.2.1.2 Area-based semantic mapping

In this category, researchers are focusing on labeling an area on the map with room name or other properties. Rottmann and Mozos *et al.* [11, 12] proposed a supervised learning approach to classify indoor places with different functional categories such as corridors, kitchens, offices, or seminar rooms. Given range and visual data, they use Adaboost along with associative Markov networks to label indoor environment.

In [24], Pronobis *et al.* developed a system that combines multimodal cues (i.e. camera and laser) by Support Vector Machine (SVM) to classify areas into classes such as corridor, offices, and kitchen.

Extended from room categories, other attributes can also be embedded in the map. Wolf *et al.* [14] built a system to map terrain types (*e.g.* navigability) and activities (*e.g.* occupancy of dynamic entities) of outdoor environment.

1.2.1.3 Semantic mapping with conceptualization

Vasudevan *et al.* [15, 16] proposed a conceptualization of objects using a probabilistic object graph representation describing the relationship among *within*, *interaction*, and *connected-through-door*. This representation is then used in place classification and place recognition.

Zender *et al.* [13] developed a system for creating conceptual representation, which is composed of four layers. The layers from low to high include: metric map, navigation map, topological map, and conceptual map. Conceptual map is a human-level understanding of the environment which consists of the relationships of *is-a* and *has-a* between the concepts and instances of place and object. They claimed that these relationships are more suitable for a situated dialogue between robots and human.

In [14], Galindo *et al.* proposed a system that utilizes a hybrid map with spatial relationship and semantic knowledge relationship (*e.g. is-a*, *has-a*) for task planning. They argued that robot's task planning can be more efficient by deducing useful knowledge from semantic structure (*e.g.* to find fridge, first go to the kitchen; or seeing fridge and then know now one is in a kitchen) and by discarding irrelevant instances during planning.

To conclude, the above mentioned semantic mapping techniques only provide information for task planning, manipulation, or navigation, not for robots to understand human-context and to infer appropriate behavior in human-robot interaction. Also, they mostly consider static context for each area, and ignore the fact that context involved with human are actually dynamic. To address these problems, we propose to embed human-context information into a spatial-temporal map for robots to efficiently determine proper behavior.

Besides the three categories mentioned above, there were only a few researches attempted to build a map with human-centric information. Gupta *et al.* [25] used 3D geometric layout of a room to predict possible human poses in the scene. They argued that the world should be understood in a human-centric way, which is similar to our idea. However, they didn’t take human internal context into account. For example, sitting on the sofa with reclined pose, people usually feel *relaxed*. The robot should behave quietly in that surrounding in order not to disturb them. Therefore, people’s interactions with objects not only have functional meaning but also influence people’s internal status.

Another closely related domain with human-centric mapping is psychogeography. This was defined in 1955 by Guy Debord [26] as "the study of the precise laws and specific effects of the geographical environment, …, on the emotions and behavior of individuals." In most related work the building process were manual, because human status is not easily detected by machine measurement. Recently, Nold *et al.* [27] proposed a pioneering system to record people's arousal on geographic maps. They used a device that combines a biometric sensor measuring Galvanic Skin Response and a GPS (Global Positioning System) to gather data of emotional arousal tagged with global location. The collected data were plotted on a geographic map. Their aim is very similar to ours. However, since their work was only used for self-emotion understanding, and for artistic purpose, no principle automatic mapping method was proposed.

1.2.2 Human context estimation

For human status sensing, a widely studied area is emotion recognition and activity recognition using human pose, voice, or physiological signal.

Coulson [28] studied the relation between human pose and emotion. In their study, experiment participants were asked to rate computer-generated mannequin figures with one of the six emotional attributes (i.e. anger, disgust, fear, happiness, sadness, and surprise). The result shows that static body posture offers a reliable cue concerning emotion. In [29], Kapur *et al.* proposed an emotion recognition system using body skeletal movements. In [30], Sebe and Huang proposed a probabilistic graphical model to fuse multimodal cues (*e.g.* facial, voice, and physiological signal) for emotion recognition. In [31], Baltrusaitis *et al.* developed a real-time emotion recognition system, which includes facial action units, head gestures and shoulder gestures. For a complete survey of emotion recognition, please refer to [19].

In this thesis our goal is slightly different from traditional emotion recognition. The general emotion categories (anger, disgust, fear, happiness, sadness, and surprise) are not very suitable for office environment. People in the office usually do not have such dramatic emotional changes. Instead, we define a new set of categories of human status in office: *concentrating*, *tired*, *relaxed*, *resting*, *social*, and *neutral*.

On the other hand, activity recognition is a wide range of study. Yao *et al.* [32] inspired our work in that activity is related not only to human pose but objects. Lan *et al.* [33] showed that group activities are also important in activity recognition. This inspired us to incorporate the concept of human-human interaction in activity recognition. These two aspects were largely unexplored in the literature of emotion recognition. Here, we do not provide exhaustive references for activity recognition. Readers can refer to Poppe's a comprehensive survey [34] on action recognition using vision, and Laptev and Mori's tutorial [18] on statistical and structural recognition of human actions.

1.3 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we describe how we estimate human-context from robot vision, using human-object interaction, human pose, and human-human interaction. Chapter 3 presents the overall mapping framework. We explain how to build a Historical map for describing human context prior. Also, the fusion of current observation and prior information for human context estimation is described. This chapter also describes the service decision making based on the inferred Human-Context Instant map. In Chapter 4, we present the simulations and experimental results. Finally, Chapter 5 summarizes the whole thesis.

Chapter 2

Human Context from Observation

2.1 Human Context in Office Environment

We believe that robot services would be more human if robots can understand human-context. However, human-context analysis is a complicated and abstract procedure in engineering field. Interestingly, in the field of drama, there are some directions for analyzing drama by examining several *dramatic elements* [35, 36]. And drama, as we believe, is a kind of simulation of real life situation, so their ideas can be borrowed to gives us some guidelines for understanding human context. In [36], dramatic elements involve the relationships between people, the relationships between people and the relationships between people and environments. In [35], the author suggests that *symbols* such as props, and gestures are powerful elements that indicate the context. These elements are the guideline for understanding human context in this work. So these concepts lead us to the design of several features in the rest of this chapter for human-context reasoning. However, here we do not seek to completely understand human context, which is beyond the scope of this thesis. Instead, we aim to provide robots a summary of human context so as to deliver context-aware services. In natural language processing, there is a similar of Probabilistic Latent semantic analysis (pLSA) [37], in which a set of concepts relating the documents and terms are produced. To achieve it, one key issue is to have an effective representation to summarize human context.

In terms of representation, there are two popular methods: one is categorical, another is parametric. In categorical representation, adjectives that describe human context are first enumerated and clustered into groups according to their meanings. When a situation or observation is given, it will be classified into the closest group. In parametric representation, a continuous space is defined with each axis describing a specific dimension. For example, to describe human emotions, Valence-Arousal space is frequently used. Valence denotes the degree of positivity or negativity (*e.g.* happiness versus miserable), whereas arousal indicates the intensity of the emotion (*e.g.* astonished versus bored).

In this thesis, we choose categorical representation to represent human context for two reasons. First, because our goal is to make appropriate decision on delivering services, for behavior designers it is easier to design the service for certain categories. If a continuous space is used, there might be regions with which are hard to associate services. Second, discretized categories are also easier for specifying their relationship with evidence. On the other hand, in this thesis we only focus on context in office environment, because studying the general contexts requires great effort, which is beyond the scope of this research work.

Therefore, to define categories for robot servicing in office environment, a questionnaire was designed to collect people’s thinking about office context. Three questions were asked:

1) What kind of human context do you think robots should know in office environment? Use an adjective, or a sentence to describe. Please list at least three kinds.
2) For each context above, as a human, how can you tell if people are in such context?
3) In each context you filled in above, what kind of behavior do you think the robot should/shouldn’t behave?

Six people who work constantly in office environment took the questionnaire. In the result analysis, our aim is to build a simple but effective model of human context. Therefore, words for describing similar context are clustered into one category. Also, categories which lead to similar robot behaviors are also merged, because our ultimate goal is for robots to infer proper behavior. As a result, we come out with the following six categories: *concentrating*, *tired*, *relaxed*, *napping*, *social*, and *neutral*. For each context, three questions are essential in our study: a) what is the definition of this context? b) What cues are typical for recognizing this context? c) What behavior/service is appropriate in this context? The latter two questions will be addressed in Chapter 3, whereas the first question is addressed here by text definitions. These definitions are adapted from Longman dictionary [38] as follows:

1) Concentrating: focusing on someone or something, usually with steady eye focus.
2) Tired: feeling that you want to sleep or rest, usually after working or meeting for a long time.
3) Relaxed: doing something enjoyable at scheduled break time or after working.

4) Napping: sleep for a short time at scheduled break time or after working.

5) Social: meeting people and forming relationships. This context usually happens in break times or social events, in which people are having light and informal conversations.

6) Neutral: not people present or situation which is none of the above.

2.2 Preliminary tools

In section 2.1 , the five categories of human context have been defined for office environment. In the rest of this chapter, we deal with the problem of inferring human context by dividing the information source into three parts: human-object interaction, human pose, and human-human interaction.

For simplicity, we define the following notations. Object labels are defined as a set L , which contains NL descriptors, $L_1, L_2, ..., L_{NL}$. Each of them represents the existence of an object category. Human poses are denoted as a set S , which contains NS descriptors. They describe the poses performed by the NS people in the region. Human-human interaction is defined as H , which contains descriptors describing the group interaction. The above observation variables denote the observation at *current* timestamp if not stated otherwise. Human context map at time t is defined as $E^{(t)} = \left\{E_i^{(t)}\right\}$, where i denotes the grid cell index, and $E_i^{(t)}$ has the domain $\left\langle concentrating, tired, relaxed, social, neutral \right\rangle$. In probabilistic notation, the methods in this section aim to estimate $P(E \mid L), P(E \mid S)$, and $P(E \mid H)$ given a scene in RGB-D image.

2.2.1 Nearby region (NR)

Given the location (x, y) where we want to estimate human context, the evidences used will be constrained inside the region, $NR(x, y, R)$, which is defined as

$$NR(x,y,R) \triangleq \left\{ (x',y') \middle| \begin{array}{l} \text{dist}((x',y'),(x,y)) \leq R), \\ \text{no\_obstacle\_in}(\text{seg}((x',y'),(x,y))) \end{array} \right\}, \tag{2.1}$$

where $\text{dist}(\cdot)$ calculates the Euclidean distance between two points, and $\text{seg}(A,B)$ defines a line segment by points A and B. In the following, this region will be referred to as nearby region (NR). The $\text{no\_obstacle\_in(seg)}$ operation, is calculated by ray tracing in 2D occupancy grid map to examine if there are obstacles along the line segment. Therefore, when using observation to update human-context grid map, each cell, referred to by its centroid as (x, y), will be enumerated and updated using the observed information in NR.

2.2.2 Proximity index (PI)

Proximity index (PI) is an attribute which estimates how strong a human is engaged in the interaction with another person or an object. PI is defined as a real number, ranging from 0 (the weakest) to 1 (the strongest). We construct the formula of calculating PI with two criteria. First, the shorter the distance between humans, the stronger the connection. Second, if the people are facing toward each other, the connection is more intense and decreases as facing direction of a person is deviating from that toward another person. So, for two individuals p_i and p_j, having their poses being $(x_{p_i}, y_{p_i}, \theta_{p_i})$, and $(x_{p_j}, y_{p_j}, \theta_{p_j})$ respectively, the angular distance can be defined as

$$\begin{aligned} &\text{ang\_dist}(p_i, p_j) = \\ &\quad \left| \text{wrap\_to\_pi}(\text{atan2}(y_{p_i} - y_{p_j}, x_{p_i} - x_{p_j}) - \theta_{p_j}) \right|, \end{aligned} \tag{2.2}$$

where $\text{wrap\_to\_pi}(\cdot)$ wraps angle in radians to $[-\pi \quad \pi]$. Notice that the angular distance is not a symmetric distance function. By combining the angular distance function and Euclidean distance function, the proximity index is calculated by the following formula:

$$\begin{aligned} \text{prox}(p_i, p_j) = \eta \mathcal{N}(&\text{dist}(p_i, p_j); \sigma_d) \\ &\times \mathcal{N}(\text{ang\_dist}(p_i, p_j); \sigma_a) \\ &\times \mathcal{N}(\text{ang\_dist}(p_j, p_i); \sigma_a), \end{aligned} \tag{2.3}$$

where $\mathcal{N}(x;\sigma)$ is a zero-mean normal distribution with variance σ, $\text{dist}(\cdot)$ calculates the Euclidean distance of the two people, and η is a normalization term making the maximum proximity 1. The visualization of proximity index function is plotted in Fig. 2-1 (b).

In this work, proximity index is also applied in evaluating the degree of human's attention on objects which have clear orientation relative to the potential user. For example, people face toward the display panel when they are using monitors. We can let p_i be the user, and p_j be the object to apply the proximity function. In some cases, objects do not have clear orientation, *e.g.* a bottle of beer. We can apply a degenerated form for estimation:

$$\begin{aligned} \text{prox}(p_i, p_j) = \eta \mathcal{N}(&\text{dist}(p_i, p_j); \sigma_d) \\ &\times \mathcal{N}(\text{ang\_dist}(p_j, p_i); \sigma_a), \end{aligned} \tag{2.4}$$

where the notations are the same as those for the original function.

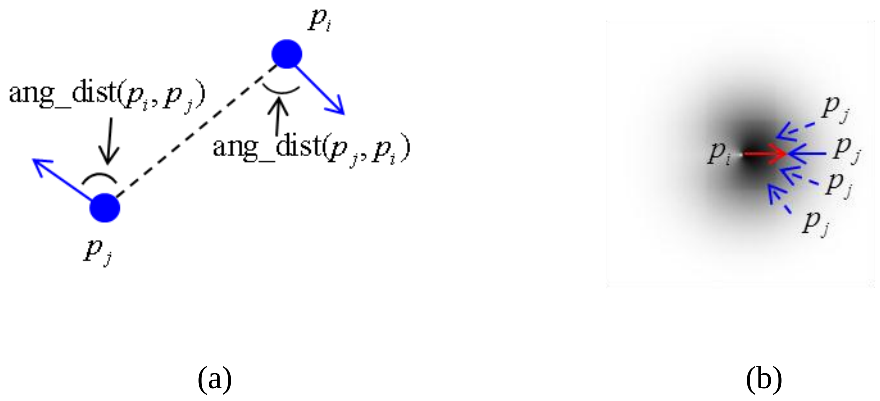


Fig. 2-1. The proximity index function.

(a) Illustration of angular distance, which is negatively correlated to the degree of facial contact. The two arrows denote the poses of two individuals. (b) A visualization of proximity function. A person p_i (shown in red arrow) is located at the center and facing right. The darkness shows the proximity index of the two when another person p_j (shown in blue arrow) is located at various locations and facing toward the person of red arrow. The value decreases as the angular distance or Euclidean distance increases.

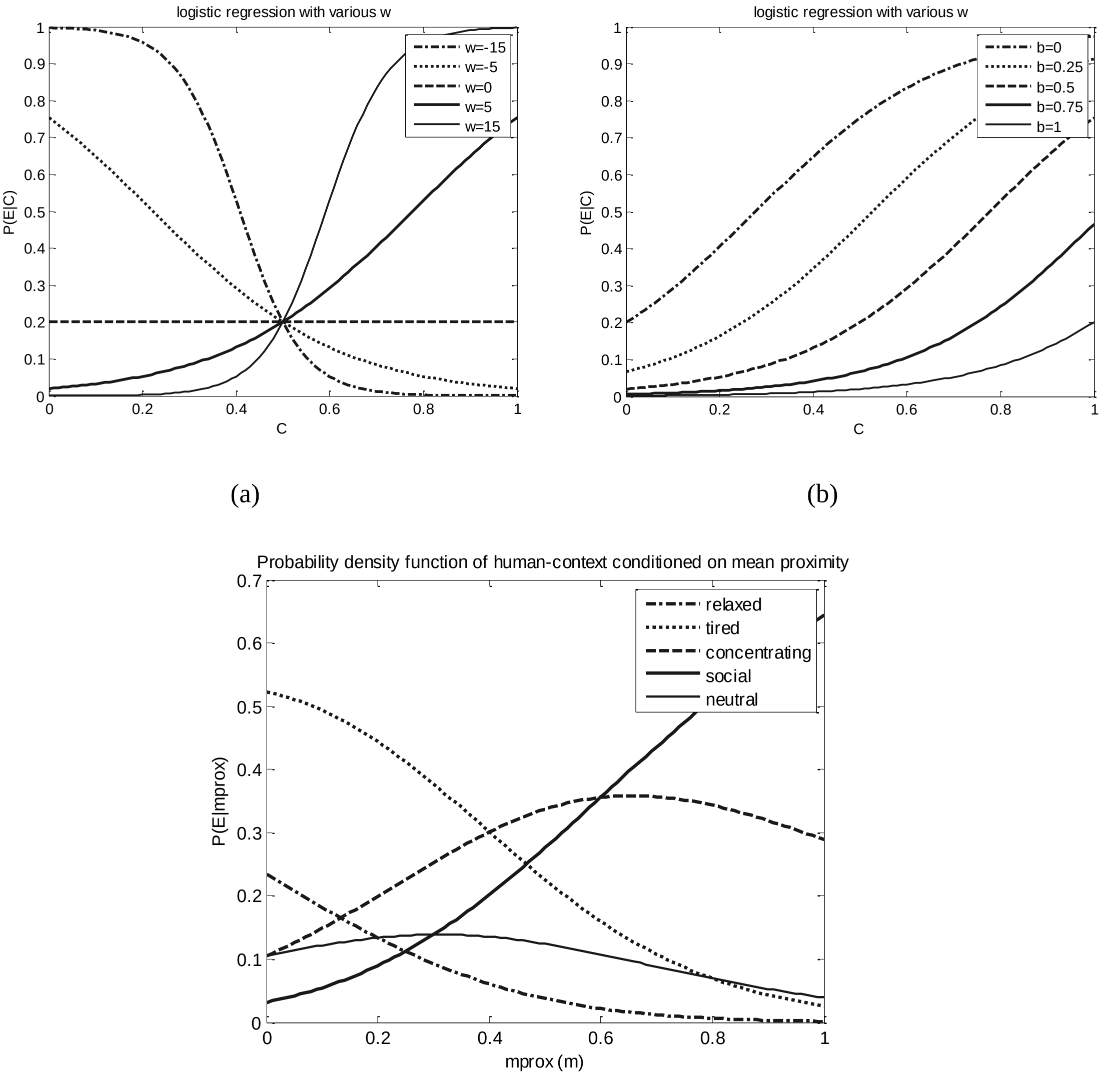
2.2.3 Logistic model

Given an evidence C, which is a continuous random variable from one of the information sources of object labels (L), human pose (S), or human-to-human interaction (H), we model $P(E|C)$ using logistic model [39]. Here, C will be represented as a value indicating how strong the evidence in that source. Two criteria made us choose logistic model. First, we assume that the intensity of certain evidence is proportionate to the occurrence of certain human context. Second, as the intensity of the evidence grows, the influence rate decreases, which conforms to the concept of marginal utility [40]. Therefore, logistic model is chosen because it meets these two criteria.

So, $P(E \mid C)$ is defined as a generalized logistic model given by

$$P(E = e \mid C) = \frac{\exp(w_{e,C} \times (C - b_{e,C}))}{\sum_{e'} \exp(w_{e',C} \times (C - b_{e',C}))} \tag{2.5}$$

For each pair of human-context (E) and evidence (C), we have to specify the parameters $w_{e,C}, b_{e,C}$ to meet the realistic situation (See Fig. 2-2 (a) (b)). Intuitively, the larger $w_{e,C}$ implies the stronger positive correlation, and vice versa. When $w_{e,C} = 0$, there is no correlation between the evidence and the human context. However, the bias parameter, $b_{e,C}$, suggests a decision threshold. Statistical learning can be applied to set the parameters from collected data using maximum likelihood. An example of using evidence from mean-proximity is shown in Fig. 2-2 (c). Mean-proximity measures how strong people are interacting based on proximity and will be detailed in Section 2.5 .



(a)

(b)

(c)

Fig. 2-2. Visualization of logistic function.

(a) Logistic model with various w, and $b = 0.5$ (b) Logistic model with various b, and fixed w (c) Probability density function of human-context conditioned on mean-proximity.

2.3 Human-Object interaction

2.3.1 RGB-D Object Recognition using Hierarchical Sparse Descriptor

In this section, we present how to obtain object labels from RGB-D images as an intermediate observation source for our Human-Context Instant mapping. Our proposed Hierarchical Sparse Shape Descriptor (HSSD) is described first. Then, we explain how to fuse multiple channels of information, i.e. color and depth, from RGB-D images.

The hierarchical representation learning contains several layers. In each layer the process is similar to a function that maps input data to output. And the output will be fed into the function of the next layer. Each layer consists of three components: sparse coding, spatial pooling, and local grouping. The system overview is shown in Fig. 2-3, and we discuss each part in detail as follows.

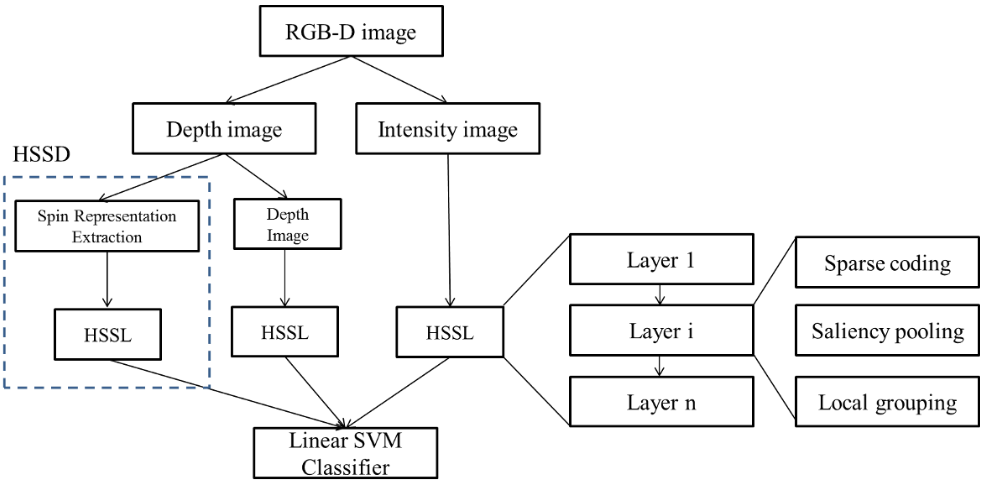
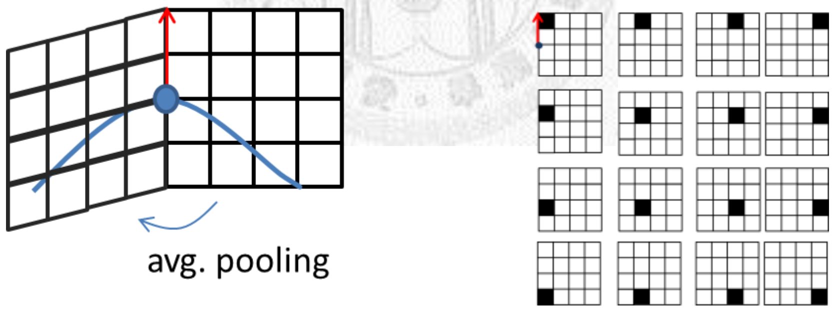


Fig. 2-3. Hierarchical Sparse Descriptor System architecture.

2.3.1.1 Spin Representation Extraction

Traditional sparse representation learning only focuses on 2D images without taking into account the physical shape information [41, 42]. One challenge of describing shape information is to achieve rotational invariance. Here, we use filter bank (dictionary) and pooling to describe the spin image and integrate it into the learning framework.

To achieve rotational invariance, the 3D descriptor must be able to align the coordinate of each feature when the local point cloud rotates. The spin image [43] utilizes local normal direction to align the first axis. Afterwards, only one degree of freedom left to rotate is along the normal. Thus, the spin image achieves invariance by making a histogram of filter response along the normal. (See Fig. 2-4)



(a) (b)

Fig. 2-4. An analogy between spin image extraction process and filtering-pooling framework. Suppose 4×4 spin images are used. (a) The normal is calculated for the filter to work on. The filter-bank is composed of 16 patterns for grids of 4x4 as shown in (b). Each filter contains only one black area that will respond to the presence of points. Average pooling is done in a spinning manner to compute the histogram. The red arrow in (b) indicates the normal direction of the spin image filters.

Next, we utilize the techniques of sparse coding to automatically find out patterns which can describe natural shapes most effectively. This is in contrast to that of manually defining shapes like plane, cylinder, and edge [44]. To provide sparse coding with rotationally invariant input, the spin image $\mathbf{P}^{spin} \in \mathbb{R}^{w_s \times w_s}$ is computed with physical radius r_s *cm* at each sampled point. The parameters are chosen as w_s = 16 and r_s = 5 if not stated otherwise. The shape signal is taken as $\mathbf{x} = \mathbf{P}^{spin}(\cdot)$ [1] for sparse coding.

2.3.1.2 Sparse coding

To find a compact shape representation, a set of bases are learned so that they can reconstruct the input signal using the weighted sum. The corresponding weight coefficient is the coding result **s**. The bases can be represented as a set of *d*-dimensional vectors, a.k.a dictionary, $\mathbf{B} = [b_1, b_2, ..., b_k] \in \mathbb{R}^{d \times k}$. Given a dictionary, the sparse code of an input signal $\mathbf{x} \in \mathbb{R}^d$ is computed by solving the following minimization problem,

$$\arg\min_{\mathbf{s}} \frac{1}{2} \|\mathbf{x} - \mathbf{B}\mathbf{s}\|_2^2 + \gamma \|\mathbf{s}\|_1 , \tag{2.6}$$

where $\|.\|_n$ denotes ℓ_n-norm, and γ denotes the regularization parameter. The first term is to minimize reconstruction error, whereas the second is to minimize the number of nonzero coefficient used to reconstruct the observed signal $\mathbf{x}$. The reason of using ℓ_1 regularization instead of ℓ_0 is that solving ℓ_0 regularization is an NP-hard

[1] $(\cdot)$ is an operator to reshape a matrix into a long vector.

problem. So, in the sparse coding literature, researchers use ℓ_1 regularization to approximate the sparseness calculated by ℓ_0-norm.

In coding phase, we expect the result to be stable, *i.e.* minor changes have small effect on **s**. To improve the stability, an additional ℓ_2-norm regularization is introduced to form an elastic net problem [45]:

$$\arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{x}-\mathbf{B}\mathbf{s}\|_2^2 + \gamma\|\mathbf{s}\|_1 + \frac{\lambda}{2}\|\mathbf{s}\|_2^2. \tag{2.7}$$

This problem can be reformulated into a quadratic form and solved using coordinate decent algorithm [46]. On the other hand, finding the most suitable dictionary to represent a set of data can be useful. One idea is to solve them simultaneously to achieve the least reconstruction error and the sparsest representation for a set of data randomly sampled from the input as:

$$\arg\min_{\mathbf{B},\mathbf{s}_i} \sum_{i=1}^{n} \frac{1}{2}\|\mathbf{x}_i-\mathbf{B}\mathbf{s}_i\|^2 + \gamma\|\mathbf{s}_i\|_1 + \frac{\lambda}{2}\|\mathbf{s}_i\|_2^2. \tag{2.8}$$

Note that the objective function is not convex if both **B** and $\mathbf{s}_i$ are optimized at the same time. Therefore, we iteratively update dictionary **B** with fixed $\mathbf{s}_i$, and update $\mathbf{s}_i$ with fixed **B**. The shape dictionary learned from RGB-D dataset [47] is shown in Fig. 2-5.

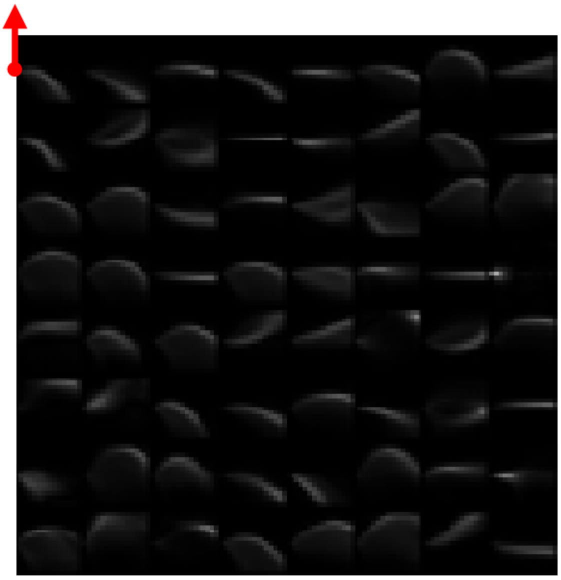


Fig. 2-5. The shape dictionary learned by calculating spin images.
The image contains 64 code words, each being of size 16×16. The red arrow indicates the normal direction of the first spin image.

2.3.1.3 Spatial pooling

In this component, statistical functions are used to combine the sparse codes in a working area into one descriptor. Functions typically used are max and average operations:

$$\text{Average-pooling:}\quad \mathbf{z} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{s}_i\ , \tag{2.9}$$

$$\text{Max-pooling:}\quad \mathbf{z} = \max_{i=1..M}\{|\mathbf{s}_i|\}, \tag{2.10}$$

where M is the number of $\mathbf{s}_i$'s in the working windows. By making a statistics, features are allowed to have translational invariance in the working window. In the literature, Boureau has empirically [48] and theoretically [49] shown that max-pooling is more robust to noise. Also, the idea from [50] uses saliency pooling, which applies biological saliency map [51] to raise the weight of the sparse code that describes foreground object, *i.e.*,

$$\text{Saliency-pooling:} \quad \mathbf{z} = \max_{i=1..M}\{w_i\,|\mathbf{s}_i|\}\,. \tag{2.11}$$

Note that pooling features can be done in different scales. For example, for spatial pyramid matching, the working area is divided into 1×1, 2×2 and 4×4 sub-spaces. Then, pooling operation is applied in each sub-space and the resulting 31 descriptors are concatenated to form the final descriptor. By doing so, spatial relationship can be retained.

2.3.1.4 Local grouping

After forming locally translation-invariant descriptors, the nearby features are grouped to construct a higher-level descriptor that can represent more complex structures. We can see this as a way to describe co-occurrence and spatial relationship of the local parts of a larger structure. Fig. 2-6 illustrates how the grouping operation is performed.

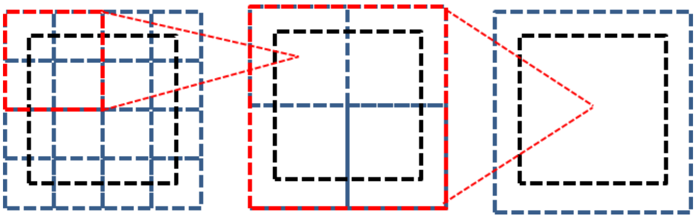


Fig. 2-6. Local working area grouping.

Local working area (depicted in dashed line) grows larger from left to right. The contour described grows from small line segments to corners and to a square. Therefore, by combining the contours of the local region, a gradually higher-level representation is formed.

2.3.1.5 Fusion of Multi-channel 2D image

It is important to note that **x** may contain multiple channels, *e.g.* for RGB-D images. Given a *f*-channel image patch $\mathbf{P} \in \mathbb{R}^{w \times w \times f}$, we form the observation signal by $\mathbf{x} = \mathbf{P}(\cdot)$. We chose $w = 8$ if not stated otherwise. Given **x**, the following the steps are described in Section 2.3.1.2-2.3.1.4. Hierarchical descriptor with Sparsity, Saliency, and Locality (HSSL) [50] is computed. From our previous work [52], we obtain the best performance by combining feature vector of HSSD with that of

Depth HSSL: one channel computed from RGB image, and

Intensity HSSL: one channel computed from RGB image,

and learn the weight by linear Support Vector Machine (SVM). Therefore, we use this configuration for our object recognition component.

2.3.2 Object detection

To detect small objects, such as calculators, binders, we first construct a recognition system using the method described in Section 2.3.1 with linear SVM [52]. Then sliding window approach is used to scan through the image for detection. For larger plane structure such as cubicles in office environment, detection method based on sliding window is not effective. Thus, we use the plane fitting techniques, and then apply rule-based decision to classify them into one of the object categories.

2.3.3 From Objects to Human Context

Here, we describe how objects provide information for estimating human-context. Although objects occur in a scene relates to human context, the interaction between human and objects can provide more precise information about human context. For calculating the degree of a person's engagement to the interaction an object L_i, human proximity to object is taken into account. That is, the higher the proximity to that object, the stronger the interaction. Therefore, L_i is defined as:

$$L_i = \sum_{j=1}^{N_{L_i}} \max_{\text{person in NR}} (\text{prox}(\text{object instance } j \text{ of class } i \text{ , person})) , \qquad (2.12)$$

where N_{L_i} is the number of object instance of class i which appeared in the scene, and NR (nearby region) is defined in Section 2.2.1. If there is no person in the NR, the prox(.) function returns zero. Then, for each pair of object category and human-context, we specify the parameters in logistic model to describe $P(E \mid L_i)$, where i is the index of object class. We associate objects to the context categories as shown in Table 2-1

Context	Related object (large)	Related object (small)
Concentrating	meeting table desk projector screen cubicle	monitor, laptop, keyboard calculator notebook, binder
Tired	meeting table desk projector screen cubicle	monitor, laptop, keyboard calculator notebook, binder
Relaxed	sofa	fruit (apple, banana) drink (soda, beer)
Napping	N/A	N/A
Social	sofa	drink (soda, beer)
Neutral	N/A	N/A

Table 2-1. Objects related with human-context categories.

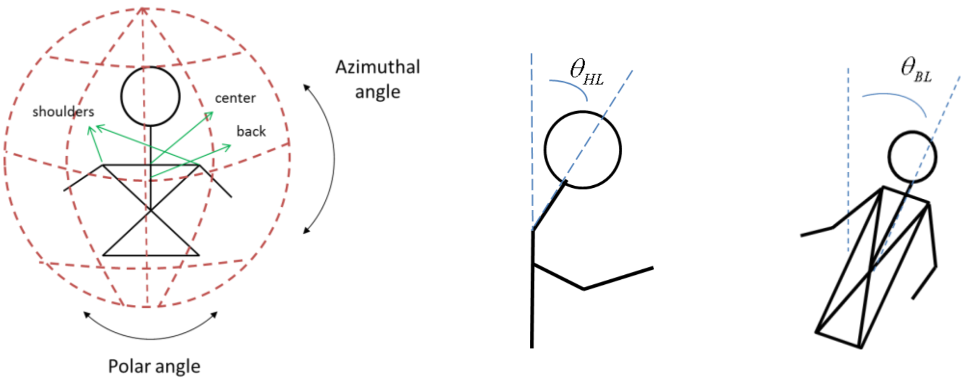
2.4 Human Pose and Motion

2.4.1 From human pose to human context

In human-context estimation, human pose, especially, upper-body pose, and motion provides rather informative cues. Here, we propose four features that describe 1) hand position relative to body, 2) head leaning, 3) back leaning, 4) histogram of gradient, and 5) motion speed.

Due to the recent development of real-time human-pose estimation by Sutton *et al.* [53], the first three features are based on the skeleton extracted from depth image. First, the hand position conveys several human statuses. For example, people propping their chin in their hands shows tiredness, and putting hands behind their head shows relaxing. Here, we record the hand position by specifying a spherical coordinate on human body. The center of the coordinate is at the middle of two joints of shoulders, the vertical axis is aligned with the back, and the line connecting two shoulder joints aligns with the horizontal axis. An illustration is shown in Fig. 2-7 (a). The hand position on the spherical axis can be specified by (r, θ_p, θ_a): radial distance r, polar angle θ_p, and azimuthal angle θ_a. The value of r is quantized into 2 intervals, θ_p into 4 bins, and θ_a into 8 bins.

Head leaning is defined as the signed angle between neck and back. We take forward leaning as positive (see Fig. 2-7 (b)). A typical example is that people tend to lean forward as they are concentrating on something. On the other hand, we define back leaning to be the signed angle between back and global vertical axis (see Fig. 2-7 (c)). For instance, people tend to lean backward when they are relaxing.



(a) (b) (c)

Fig. 2-7. The illustration of (a) hand position (b) head leaning (c) back leaning.

In practice, the above three features rely on robust skeleton tracking, which is hard in the case of sitting at desk because of serious occlusion. However, this situation is very common in office environment. We have experimented on using NITE package to track the skeleton of sitting people, and the result was not good enough. Therefore, we introduce a direct image level feature: histogram of gradient (HOG). HOG was proposed by Dalal *et al.* in 2004 for human detection [54], and used for action recognition in [55, 56]. HOG feature encodes the 2D contour of a human by aggregating histograms of gradient computed in an array of local cells (16x16 pixel^2). The gradient indicates the orientation of the edge feature in a cell. Therefore, we can use HOG to encode pose information, *e.g.* head leaning and back leaning, in a holistic fashion. From our experiments, HOG feature works better in side view whereas the skeleton tracking works better in frontal view, so they are complimentary features. An example of applying HOG on a small data set for human context classification is shown in Fig. 2-8. In cross-validation, accuracies of 97.5% and of 98% have been achieved on intensity and depth image, respectively. At this point the images contain only one person data and from only one viewpoint, in experiment section, a more comprehensive experiment will be conducted.

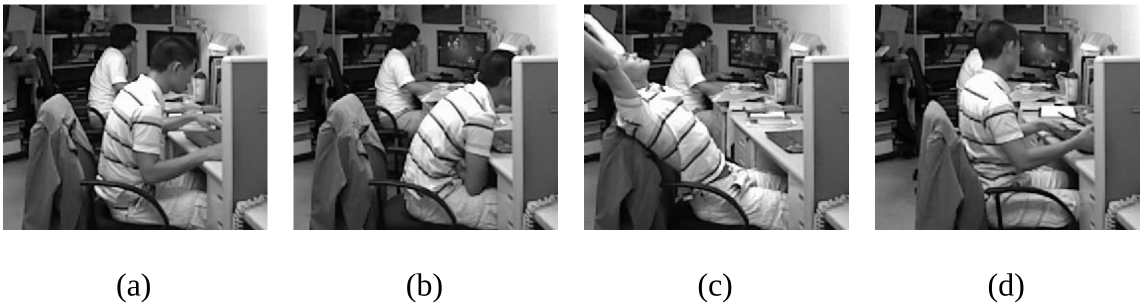


Fig. 2-8. A small dataset of 200 RGB-D images.

The collected images were divided into 4 of the 5 human-context categories. The category of social context was ignored at this point because a typical social scene involves multiple people. Fig. 2-8 (a-d) are examples of concentrating, tired, relaxed, and neutral.

On the other hand, human motion speed can indicate the tension of people. For example, people walk slower when they are relaxing or having social conversation. We define the mean velocity in NR as:

$$\mathrm{mve} \triangleq \mathrm{E}_{p_i \in NR}\left[\left|v_{p_i}\right|\right] \tag{2.13}$$

where $\mathrm{E}[.]$ means expectation, and $\left|v_{p_i}\right|$ is the motion speed of a person p_i.

2.4.2 Estimating Human Pose and Motion from Depth Image

We use OpenNI and NITE package [57] for detecting human in depth images and skeleton estimation. So, motion speed, hand position, head leaning and back leaning for each individual can be estimated.

2.5 Human-Human Interaction

Here, we define two types of feature in describing Human-Human interaction, 1) mean proximity; and 2) orderliness.

First, mean proximity is defined as the average of proximity values of every people in NR to the person with highest proximity.

$$\text{mprox} \triangleq \underset{p_i \text{ in } NR}{\mathrm{E}} \left[\max_{p_j \text{ in } NR} (\text{prox}(p_i, p_j)) \right], \tag{2.14}$$

where NR (nearby region) is defined in Section 2.2.1, and prox(.) calculates the proximity index defined in Section 2.2.2. Usually, people in social context or engaged in discussion, will have higher proximity. On the other hand, people in individual situation such as relaxing, proximity is lower.

Secondly, the orderliness is defined as

$$\text{ord} \triangleq C(N_{ppl}, 2) / N_{line}, \tag{2.15}$$

where C is a combination function, N_{line} is the Minimum number of lines that can fit all the points occupied by people, and N_{ppl} is the number of people in *NR*. In formal situation, *e.g.*, meeting, people tend to have regular patterns in sitting, whereas in social context, people tend to sit or stand in arbitrary pattern.

Chapter 3

Human-Context Mapping Framework and Behavior decision

In this chapter, we propose a human-context mapping framework, which is a probabilistic framework that integrates various information sources for human-context inference. A Dynamic Bayesian Network (DBN) is used as the foundation of our probabilistic framework. This framework incorporates both current observation and previous estimations for reasoning. (See Fig. 3-2)

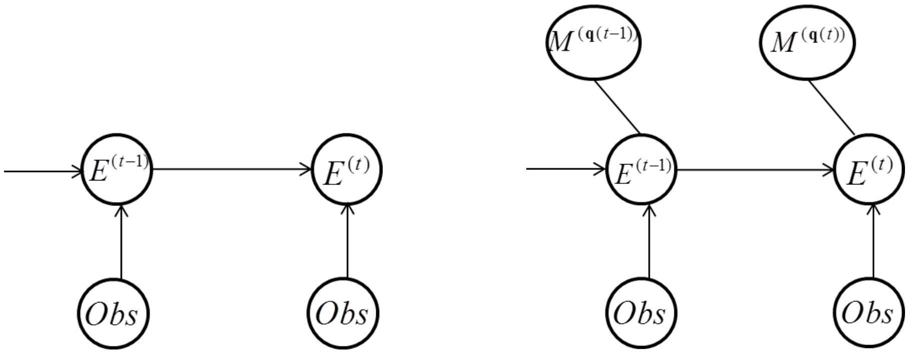
3.1 Inference from the past

In many dynamic systems, inference over time is very common, in which current state depends on the previous state. In our human-context inference, we observe the tendency that the current state is similar to the last state, so the transition probability from the previous to the current is defined as

$$P(E_i^{(t)} = e \mid E_i^{(t-1)} = f) = \begin{cases} p & \text{if } e = f \\ (1-p)/(N_E - 1) & \text{otherwise} \end{cases} \quad (2.16)$$

where $E_i^{(t)}$ is the state estimate at timestamp t and grid cell i, $p \in [0,1]$ indicates the similarity, and N_E denotes the number of human-context categories ($p = 0.9$ and $N_E = 5$ in this research). In addition, due to the nature that the state of *concentrating* tends to transition to *tired*, and *tired* state usually leads to *relaxed* state, we can increase these transition probability by ε_p and normalize it afterwards. It is also possible to learn parameters statistically instead of manual tuning.

Moreover, what we would like to point out is, in office environment, people tend to have fixed activity patterns in the periodicity of day and even week. Therefore, we propose to expand the dependency on the previous state to the past states, which are at the same time of day in yesterday or in the same time of week. To utilize this prior, we proposed a Historical map ${}^{Q}M = \{{}^{Q}M^{(1)}, {}^{Q}M^{(2)}, ..., {}^{Q}M^{(N_Q)}\}$, where N_Q is the number of local timestamp in periodicity Q. Examples of periodicity Q for normal humans are “24 hours a day” and “7 days a week”, so the maps that include these priors are specified by ${}^{Day}M$ and ${}^{Week}M$. In the case of daily period, the local timestamp only denotes time of the day (*e.g.* 8:00 AM). For weekly period, the local timestamp refers to day of the week and time of the day (*e.g.* Wednesday 8:00 AM). Other kinds of periodicity can also be incorporated if applicable. So, for each periodicity, a function that transforms the global timestamp to local timestamp should be defined, *e.g.* from 2012/5/17 15:00 to Thursday 15:00 if we are using week periodicity. The function of periodicity *Q* is noted as small letter *q*(.). Therefore, we define the transform function for day periodicity as $day(t)$, and week as $week(t)$. In practice, the prior provides information according to the conditional probabilistic tables of $P(E^{(t)} | M^{\mathbf{q}(t)})$, where $M^{(\mathbf{q}(t))} = \{{}^{Day}M^{(day(t))}, {}^{Week}M^{(week(t))}\}$ for each periodicity (See Fig. 3-1). Sometime, ${}^{Q}M^{(q(t))}$ will be written as $M^{(q(t))}$ for simplicity because function name also specifies the periodicity.



(a) (b)

Fig. 3-1. Inference over time.

(a) General conditional inference over time. (b) Proposed conditional inference with periodic history prior. Obs indicates the observations.

3.2 Historical grid mapping

Given the current estimation $\mathbf{P}(E^{(t)})$, the goal here is to update the Human-Context Historical map $\mathbf{P}(\hat{M}_i^{(t)})$. To recursively update the Historical map of Q, we use discrete Bayes filter:

$$\mathbf{P}(M_i^{q(t)}) = \eta \underbrace{\mathbf{P}(E_i^{(t)})}_{\text{update}} \sum_e \underbrace{\mathbf{P}(M_i^{q(t)} | \hat{M}_i^{q(t)} = e) P(\hat{M}_i^{q(t)} = e)}_{\text{predict}}, \tag{2.17}$$

where $\mathbf{P}(\hat{M}_i^{(t)})$, $\mathbf{P}(M_i^{q(t)})$ are respectively the outdated and updated Historical map,

γ is the learning rate, which sets the weight of the previous estimation and the current one, and η is the normalization term. The transition function $\mathbf{P}(M_i^{q(t)} \mid \hat{M}_i^{q(t)})$ is defined the same as in (2.16).

Algorithm historical_grid_mapping($\{\mathbf{P}(\hat{M}_i)\}$, $\{\mathbf{P}(G_i)\}$, $q(t)$):

```
for all periodicity $Q$
    let $q$ be the transformation function of $Q$
    for all cells $M_i$ do
        if ( $G_i$ is Unoccupied )
            $\mathbf{P}(\bar{M}_i^{q(t)}) = \sum_e \mathbf{P}(M_i^{q(t)} \mid \hat{M}_i^{q(t)} = e)\ \mathbf{P}(\hat{M}_i^{q(t)} = e)$
            $\mathbf{P}(M_i^{q(t)}) = \eta\ \mathbf{P}(\bar{M}_i^{q(t)})\ \mathbf{P}(E_i^{(t)})$
        endif
    endfor
endfor
return $\{\mathbf{P}(M_i)\}$
```

Table 3-1. Historical Grid Mapping algorithm.

3.3 Information Fusion

In Chapter 2, we presented the three observation sources, object labels (*L*), human-pose (*S*), and human-human interaction (*H*). And in Sections 3.1 , 3.2 inference from the past and Historical map are presented. To fuse all these information sources, Dynamic Bayesian Network (DBN) [58] is chosen as the fusion model due to its flexibility. More importantly, DBN can probabilistically handle Label Bias Problem [58], with guaranteed global maximum likelihood convergence.

We formulate the unnormalized conditional probability as the summation of output conditional probability from every information source.

$$\begin{aligned}
&\ln P(E^{(t)} \mid L, S, H, M^{\mathbf{q}(t)}, E^{(t-1)}) \\
&= \overbrace{\sum_i \ln P(E^{(t)} \mid L_i; \theta_\psi)}^{\text{human-object interaction}} + \overbrace{\sum_i \ln P(E^{(t)} \mid S_i; \theta_\pi)}^{\text{human posture}} + \overbrace{\sum_i \ln P(E^{(t)} \mid H_i; \theta_\lambda)}^{\text{human-human interaction}} \\
&\quad + \overbrace{\ln P(E^{(t)} \mid E^{(t-1)}; \theta_\phi)}^{\text{temporal constraint}} + \overbrace{\ln P(E^{(t)} \mid M^{\mathbf{q}(t)}; \theta_\kappa)}^{\text{periodic constraint}},
\end{aligned} \tag{2.18}$$

where the first three terms are introduced in Chapter 2, and the last two terms are presented in Section 3.1 .

To make conditional probability sum to 1:

$$\begin{aligned}
&\ln P(E^{(t)} \mid L, S, H, M^{\mathbf{q}(t)}, E^{(t-1)}) \\
&= \ln P(E^{(t)} \mid L, S, H, M^{\mathbf{q}(t)}, E^{(t-1)}) - \ln Z,
\end{aligned} \tag{2.19}$$

where $Z = \sum_{e \in E} P(E^{(t)} = e \mid L, S, H, M^{\mathbf{q}(t)}, E^{(t-1)})$ is the partition function.

Moreover, based on the historical prior and transition probability, the robot can predict several steps in the future. This makes the service planning more possible to achieve global optimal, because in real cases we need to consider the motion time to reach certain goals, and some motion primitives can be conducted at the same time to achieve higher efficiency.

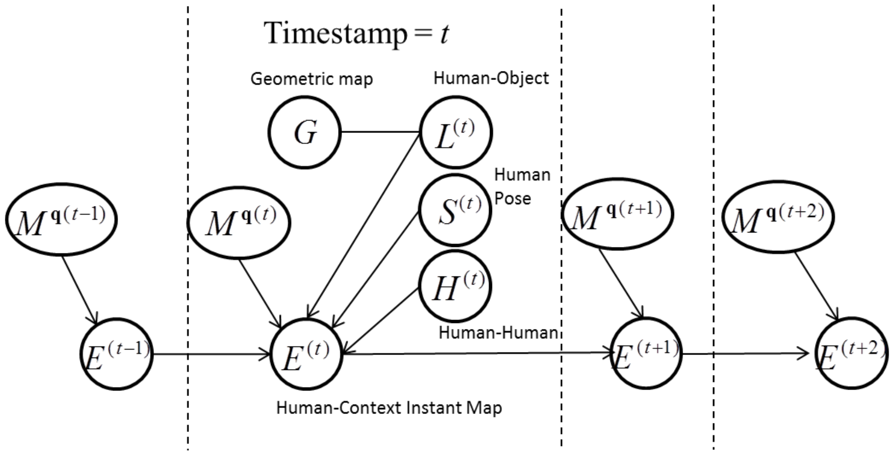


Fig. 3-2. A Dynamic Bayesian Network for Human-Context Instant map prediction.

3.4 Behavior decision in Office Environment

From the Human-Context Instant mapping, the probability of human context is obtained. Based on this information, the robot can decide their service behavior with the awareness of human context. At the same time, the probability estimated also helps the robot to decide whether it needs to acquire more information. When the uncertainty is low, it can be very sure about the human context at certain area. Otherwise, the robot may need to spend some efforts to obtain more information from observation. Here, we focus on service providing, and information gathering is reserved for future work.

In this work, behavior is composed of three channels of output. First, what service tasks and corresponding motion primitive a robot should take? Second, what is the loudness of speaker volume should be? Third, whether a robot should interrupt a person to deliver a message or not?

3.4.1 Service Task

Here we define a task as high level activity such as coffee serving, which is composed of a series of motion primitives (MPs). MPs are defined as some short motions like “moving to kitchen”, “fetching coffee cup”, “filling coffee”, and “returning to meeting room”. So, each task can be described as a finite state machine with each node being an MP. (See Fig. 3-3). To achieve the goal of a task, the sequence of MPs is dynamically generated according to robot’s current state, and the Human-Context Instant map. For example, the robot’s current location is at Office 1 and it noticed there is a social context which is occurring in Meeting room, so it wants to serve coffee to the Meeting room. Before the robot fetches coffee from the kitchen, it has to include the behavior which leads the robot to kitchen. After the coffee is fetched, the movement from kitchen to Meeting room should be planned. Therefore, the task is dynamically organized and can be presented as a function of the robot state and a goal:

$$T:\{robotstate\}\times\{goal\}\rightarrow\{b^{(1)},b^{(2)},...,b^{(N_T)}\}, \tag{2.20}$$

where T is task name with N_T output MPs, whereas the formats of $robotstate$ and $goal$ depend on the task.

For service task, only when the robot reaches the finish state, can it receive the reward (utility > 0). That is, if it fails at some point because of other unpredictable event, it won’t receive the reward.

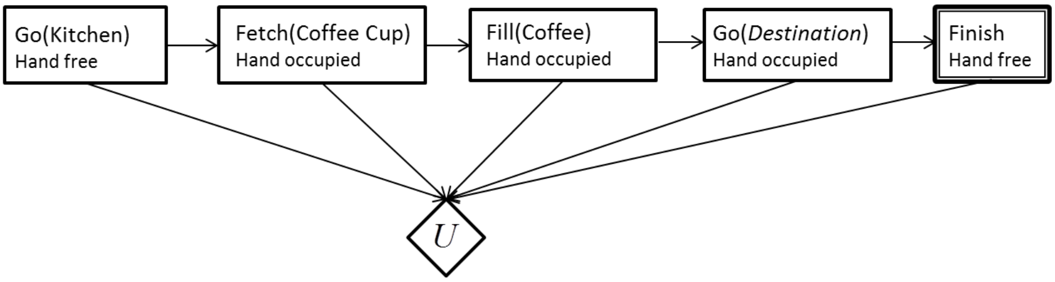


Fig. 3-3. Using finite state machine to construct a service task, serve coffee, from motion primitives.

3.4.2 Motion Planning

Given the prediction of Human-Context Instant map $\{E^{(t)}, E^{(t+1)}, ..., E^{(t+N_p)}\}$, with N_p steps look ahead. The goal is to find a sequence of MPs $\mathbf{b} = \{b^{(t)}, b^{(t+1)}, ..., b^{(t+N_p)}\}$, which leads to the highest utility score in expectation. The graphical model is shown in Fig. 3-4.

We formulate the score of a sequence of behaviors as :

$$score(b^{(t)}, ..., b^{(t+N_p)}) = \sum_{i=0}^{N_p} P(b^{(t+i)} \mid b^{(t+i-1)}) U(b^{(t+i)}, E^{(t+i)}), \tag{2.21}$$

where $U(b)$ is the utility function that evaluate the MP b .So, the goal is to find $\mathbf{b}$ that maximize the target score function.

$$\arg\max_{\mathbf{b}} \{score(\mathbf{b})\} \tag{2.22}$$

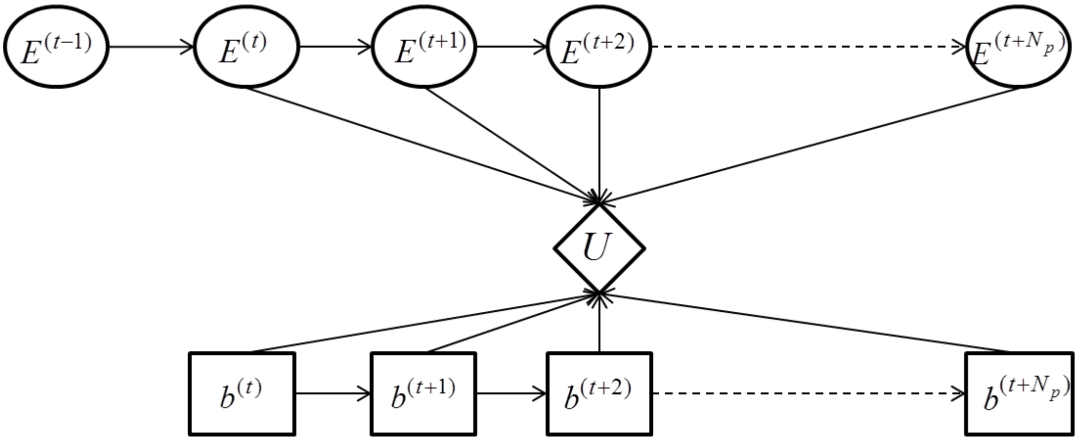


Fig. 3-4. Graphical model of decision network.

To solve this problem, we apply depth-limited search. That is, the maximum depth is set to N_p, and conduct depth-first search. However, if we directly conduct searching in MP space, the complexity will be very high for a complete search. Notice that we have the constraint of tasks, which group the MPs into meaningful service tasks. Therefore, searching in task space will greatly reduce the complexity. By doing so, the search tree will have branching factor equals to number of Tasks instead of number of low-level behaviors.

3.4.3 Associating Behaviors with human context

From the previous section the abstraction of task decision problem has been formulated and a solution was provided. Remember that besides actions, the behavior also include the volume of robot's speaker volume and interruption rules.

For each context, we build a table of associated service tasks, volume, and interruption rule. The content is extracted from the result of the questionnaire mentioned in Section 2.1 . (See Table 3-2).

Context	Task ($U(T,E)$)	Volume	Interruption
Concentrating	Serve Coffee (8)	Low	Important message only
Tired	Serve Coffee (8) Book massage (7) Play music (6)	Low	Important message only
Relaxed	Play music (8)	Medium	Yes
Napping	Turn off lights (8)	Low	Emergent message only
Social	Serve snacks (8) Report news (7) Shake hands (6)	High	Yes
Neutral	Greeting (3)	Medium	Yes

Table 3-2. Behaviors associated with human-context categories.

$U(T,E)$ is zero if not specified above.

Chapter 4

Evaluation

Here, we evaluate our algorithms in three parts, 1) human-context inference from observation, 2) building Human-Context Instant map, 3) showing a robot servicing people based on the current Human-Context Instant map in real environment.

4.1 Human-context inference from observation

In this work, the observation source is divided into three parts: object labels, human pose, and human-human interaction. Each part is detailed as follows.

4.1.1 Human-object interaction

4.1.1.1 Object Recognition

Dataset

The dataset we use is the first 10 object categories from the large scale RGB-D dataset proposed in [47]. The objects picked are shown in Fig. 4-1. Average accuracies and standard deviation for each experiment were obtained across 10 trials. The objects were put on a turn table and captured using a depth sensor and a higher resolution RGB camera.

We subsampled the dataset by taking every fifth frame, resulting in 6,258 RGB-D images. The point cloud captured for each view was downsampled to approximately 3000 points for fast evaluation. The testing theme is category level recognition. We follow the testing procedure described in [47]: randomly leave an object out from each category for testing and train the classifiers on all views of the remaining objects.

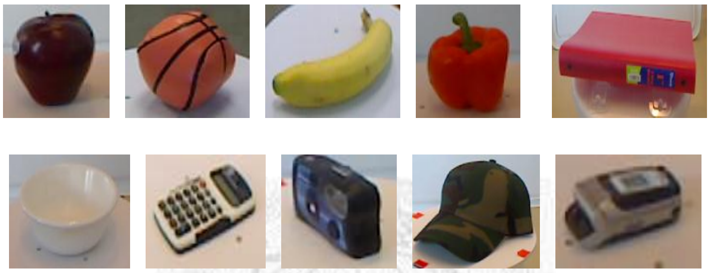


Fig. 4-1. objects categories from RGB-D dataset: apple, ball, banana, bell pepper, binder, bowl, calculator, camera, cap, and cell-phone.

Pre-processing

Before feeding raw images into the first layer, we whiten them as suggested in [59]. First, the image was resized to a fixed size of 151 pixels while maintaining the original ratio. If the image has multiple layers, the resizing is conducted independently on each channel. Second, the standard deviation of the whole image and the 9×9×d local patch is calculated (d=4 for RGB-D image, and d=16×16=256 for spin-image map). We choose the greater standard deviation as the normalizer. Then, every pixel was subtracted by the mean of the 9×9×d window and divided by the normalizer. Third, the image was zero-padded to have 143×143×d pixels.

Configuration of Learning Hierarchy

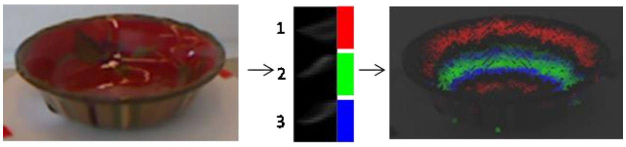
Here, the configuration is made similar to [59] for Caltech 101 dataset but with some adaptation to spin image map.

First layer: We randomly sample 200,000 8×8×*d* patches to learn a dictionary of 64 codewords. Given an image, we step over it with step size one, and compute sparse coding for each local patch. As a result, we get 136×136 64-dimensional descriptor map. The sparse code is max-pooled within each 4×4 non-overlapping window, which results in 34×34 64-dimensional descriptors. Descriptors are grouped for each pixel within 4×4 local window making a 31×31 1024-dimensional descriptor map. To reduce the dimensionality, we use PCA to project it down to 96 dimensions.

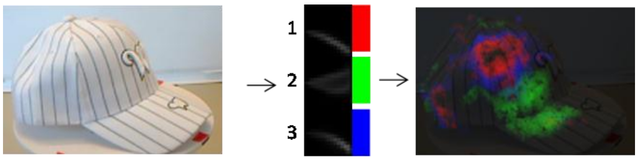
Second layer: In this layer, n_{s2} codewords are learned. We chose n_{s2}=2048 for 2D image HSD, and n_{s2}=128 for HSSD. For a given output from layer one, sparse coding will produce 31×31 ns2-dimensional descriptor map. Finally, the max-pooling is operated within 1×1, 2×2, and 4×4 subspaces of the whole image. The descriptors after max-pooling are concatenated into one single long descriptor.

Hierarchical Sparse Shape Descriptor

Figure Fig. 4-2 shows two examples that illustrate first layer sparse coding. Spin images at every point are encoded by its major shape component. For the instance of bowl, due to different curvature from its bottom to top, they are encoded by shape words that best describe it.



(a) Bowl



(b) Cap

Fig. 4-2. Two examples of first layer sparse coding, *Left*: the object image, *Middle*: the three spin image bases out of 64 that has larger response on the object, *Right*: the response of the three shape words. Red: the response of basis 1, Green: the response of basis 2, Blue: the response of basis 3.

In Table 4-1, the accuracy of category level recognition is shown. Our proposed HSSD has comparable performance with directly applying HSSL on depth images, which encodes 2D contour. More importantly, when we combined Depth HSSL with HSSD, the performance increases, which shows HSSD can compensate depth image with physical shape information. We combined different cues by concatenating the feature vectors and using linear SVM [60] as the classifier. We compared our HSSD with VFH [21], which encodes viewpoint and geometry cues using FPFH [20] of object point clouds. Although the spin representation we applied does not include statistics of normal differences as in FPFH, by learning sparse representation and hierarchical structure, our HSSD outperforms VFH by 13%.

Feature	Accuracy(%)
Intensity HSSL	90.7±4.8
HSSD	84.8±4.8
Depth HSSL	85.7±4.0
HSSD + Depth HSSL	91.3±5.4
VFH [21]	71.5±2.6
Intensity + Depth HSSL	95.5±3.4
HSSD + Intensity + Depth HSSL	96.9±2.9

Table 4-1. Accuracies (%) of several descriptors on the RGB-D10 object dataset.

4.1.1.2 Object Detection

The above result is for a formal testing of our recognition algorithm, so we used the published RGB-D dataset in order to compare the recognition performance. However, for office environment, we collected another dataset, which includes monitors, bottles, sofa, tables, chairs.

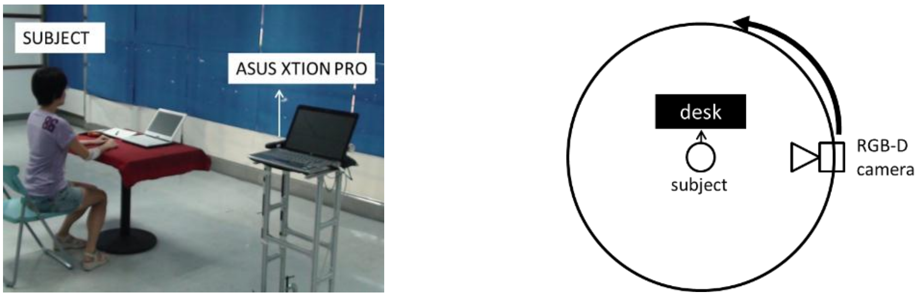
Although our proposed recognition method is accurate but the current implementation using MATLAB requires much time (about 3.7sec) to classify an object in a cropped image. So, using exhaustive sliding window approach with this method for detection is impractical. Therefore, we used a more simple but efficient detection algorithm [61] to propose detection hypotheses, and then used our recognition to validate each proposal. By tuning the threshold of the detection algorithm for proposal, we can tradeoff between computation time and detection performance.

4.1.2 Human-Pose

4.1.2.1 RGB-D image collection

In order to build a discriminative classifier for human-context recognition using RGB-D images, a set of training images were collected. A piece of sampled data is a human RGB-D image with a label of human-context category.

To make the data as complete as possible, human poses are divided into two cases, namely, sitting and standing. Also, images from every viewpoint were collected, because robots may observe frontal, profile, or back in natural office environment. This is different from the scenario of human-computer interaction where the testing views are mostly frontal. Therefore, in the data collection process, subjects were asked to sit at a desk, and perform natural postures related to each context category. We used a moving platform, Pioneer 3DX, mounted with ASUS Xtion Pro Live to circle the subject in order to capture images from different view angles. In Fig. 4-3(a), a subject was sitting at a desk in the center and performed posture associated to a specific human-context. A mobile robot mounted with an RGB-D camera circled the subject and captured images at a rate of 240 per cycle. In Fig. 4-3(b), Top view of the configuration. The radius of the circular path is 150 cm.

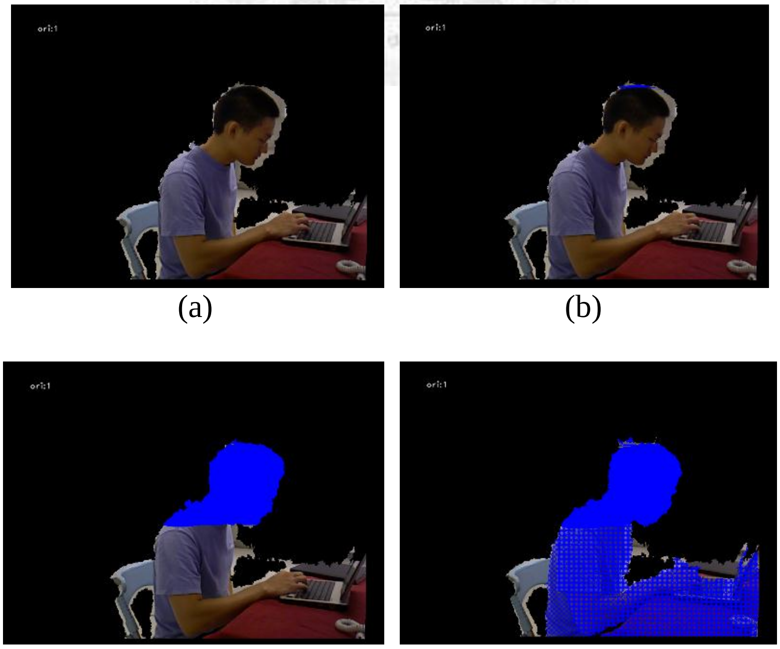


(a) (b)

Fig. 4-3. Configuration of human posture data collection.

4.1.2.2 Preprocessing

The most important step in the preprocessing stage is to automatically crop the region occupied by human. We construct a simple but effective algorithm to achieve this goal. There are mainly three steps, illustrated in Fig. 4-4. First, the background is subtracted using depth value, so only user and foreground objects are left. We set the depth threshold to 2 meter (Fig. 4-4(a)). Second, every pixel is transformed into real world coordinates as point clouds. Then, we set a horizontal scan plane of 1 cm thick, and scan from top (1.8 m above ground) to bottom (1.0 meter above ground). The scanned region is where user head may locate (Fig. 4-4(b)). Once a sufficient amount of pixels is detected at *h* meter, we push the location of pixels located between (h+ε) and (h-ε) above ground into a queue (Fig. 4-4(c)). Third, given the queue, we perform connected component algorithm to find all points of the body (Fig. 4-4(d)). Finally, we can reject a hypothesis if the portion of skin color is less than a threshold.



(a) (b)

(c) (d)

Fig. 4-4. The process of human detection.

4.1.2.3 Experiment on HOG feature

We collected data from 5 people in sitting case, and data from 6 people for standing case. Then, the data were processed as follows. First, we extracted the features representing human pose described in Section 2.4 . The data were clustered according to their view angles. Here we divided the view angles into 8 segments as shown in Fig. 4-5. Each view segment along with the context label forms a subclass. That is, there would be a total of 6 context categories × 8 segments → 48 subclasses. By doing so, the classifier can accommodate intra-class variations due to different view angles.

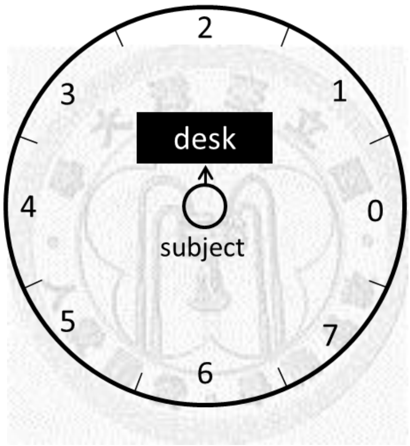
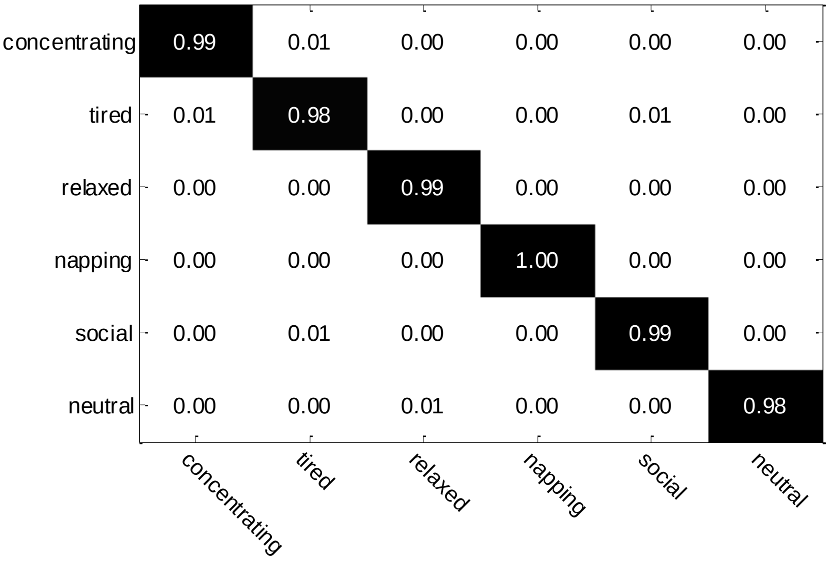


Fig. 4-5. Illustration of view angle clustering

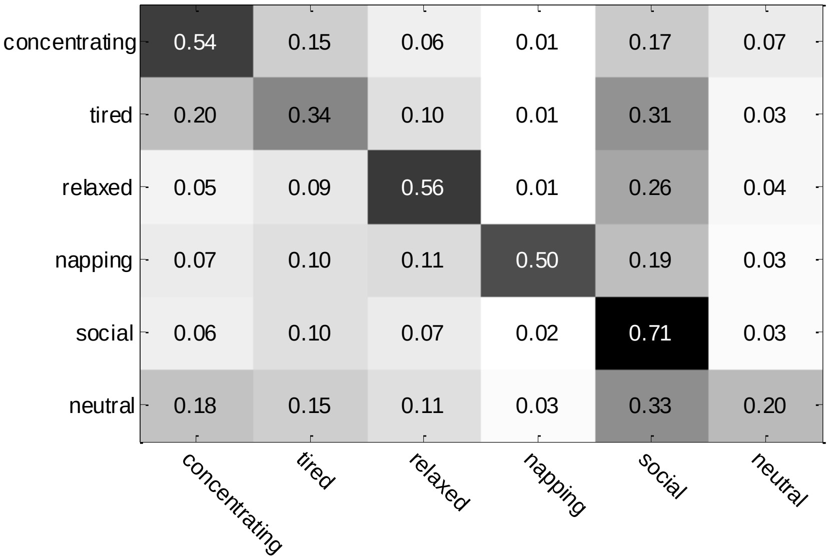
The ultimate goal is to construct a human-context classifier which can accommodate views from view angles or from users that are not trained before. Therefore, we conduct two kinds of test. One is leave-frames-out (LFO), and the other is leave-one-person-out (LOPO). In LFO, frames are randomly picked 1/5 of all frames as the validation set, and the rest are used as training set. The validation process is conducted 10 times and the accuracies are averaged. In LOPO, all data from a randomly picked user are used as a validation set, and the rest forms a training set. The accuracy of LFO is 98.68±0.14%, which is surprisingly high. Probably, this is due to intense sampling (approximately one frame for every 1.5°) and that consecutive frames are quite similar so the loss of small portions of frames does not affect the performance.

On the other hand, the accuracy of LOPO is 55.38±7.56%, which is quite lower than that of LFO. One fundamental difficulty lies in the variation of poses performed by different people. Other variations such as body configuration will also degrade the performance of HOG feature. This may resolved by collecting more comprehensive data to train the classifier.



	concentrating	tired	relaxed	napping	social	neutral
concentrating	0.99	0.01	0.00	0.00	0.00	0.00
tired	0.01	0.98	0.00	0.00	0.01	0.00
relaxed	0.00	0.00	0.99	0.00	0.00	0.00
napping	0.00	0.00	0.00	1.00	0.00	0.00
social	0.00	0.01	0.00	0.00	0.99	0.00
neutral	0.00	0.00	0.01	0.00	0.00	0.98

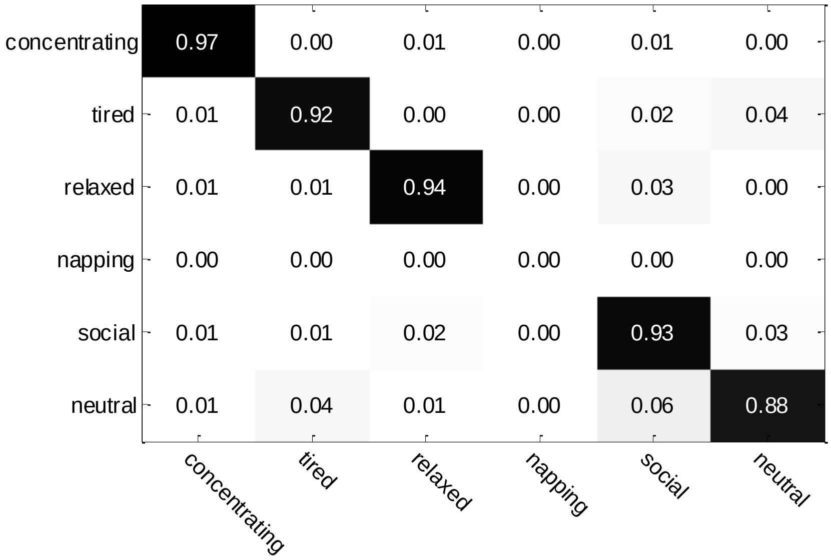
Fig. 4-6 Confusion matrix of sitting case using HOG feature, leave frames out (user dependent classification)



	concentrating	tired	relaxed	napping	social	neutral
concentrating	0.54	0.15	0.06	0.01	0.17	0.07
tired	0.20	0.34	0.10	0.01	0.31	0.03
relaxed	0.05	0.09	0.56	0.01	0.26	0.04
napping	0.07	0.10	0.11	0.50	0.19	0.03
social	0.06	0.10	0.07	0.02	0.71	0.03
neutral	0.18	0.15	0.11	0.03	0.33	0.20

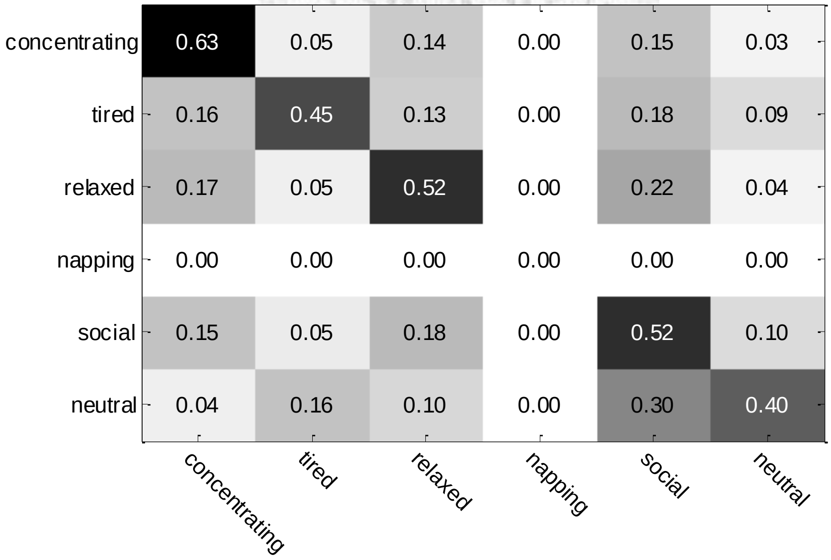
Fig. 4-7. Confusion matrix of sitting case using HOG feature, leave one person out (user independent classification)

For the case of standing, because usually people in office won’t sleep in a standing pose, so we did not capture training data for napping case. The accuracy of LFO is 93.84±0.25%, which is slightly lower than the accuracy in sitting case. This phenomenon can be attributed to the subtle difference in standing case, where people do not lean forward or backward so much as they did when sitting. On the other hand, the accuracy of LOPO is 52.13±6.43%. Still, due to inter-person variations, constructing a user independent classifier is difficult and should incorporate other cues.



	concentrating	tired	relaxed	napping	social	neutral
concentrating	0.97	0.00	0.01	0.00	0.01	0.00
tired	0.01	0.92	0.00	0.00	0.02	0.04
relaxed	0.01	0.01	0.94	0.00	0.03	0.00
napping	0.00	0.00	0.00	0.00	0.00	0.00
social	0.01	0.01	0.02	0.00	0.93	0.03
neutral	0.01	0.04	0.01	0.00	0.06	0.88

Fig. 4-8. Confusion matrix of standing case using HOG feature, leave frames out (user dependent classification)



	concentrating	tired	relaxed	napping	social	neutral
concentrating	0.63	0.05	0.14	0.00	0.15	0.03
tired	0.16	0.45	0.13	0.00	0.18	0.09
relaxed	0.17	0.05	0.52	0.00	0.22	0.04
napping	0.00	0.00	0.00	0.00	0.00	0.00
social	0.15	0.05	0.18	0.00	0.52	0.10
neutral	0.04	0.16	0.10	0.00	0.30	0.40

Fig. 4-9. Confusion matrix of standing case using HOG feature, leave one person out (user independent classification)

4.1.2.4 Experiment on skeleton-based feature

In this section, we evaluate our proposed skeleton-based feature: hand position, head leaning, back leaning for office context classification. Their definitions are in Section 2.4.1. For hand-position feature (r, θ_p, θ_a), the specific coordinate used in the following experiments is shown in Fig . Polar angle θ_p ranges over $[0° \quad 360°)$ and starts from axis *v₁*, and increase clockwise when the person is seen above head. θ_a ranges over $[-90° \quad 90°]$ has $0°$ on the plane spanned by v_1 and v_2. It increases in the direction toward head direction.

We collected a human pose dataset, in which each frame has complete upper body skeleton information using NITE package. Because the current version of the package can only capture skeleton in a frontal view and we assume that skeleton extraction system can obtain consistent skeleton from different view angle using RGB-D images, only frontal views are captured. Example views and their corresponding skeletons with extracted skeleton-based features are shown in Fig. 4-11. The information beside each view follows the format: [context name], [pose name], [left hand (LH)], [right hand (RH)], [head leaning (HL)], [back leaning (BL)], [velocity]. Lengths are in *mm*, angles are in degree, velocities are in *mm/sec*.

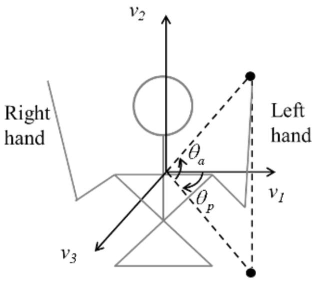


Fig. 4-10. Coordinate of hand-position feature.

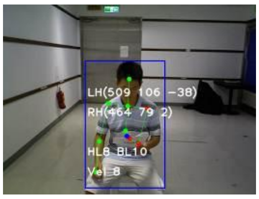
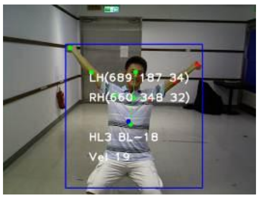
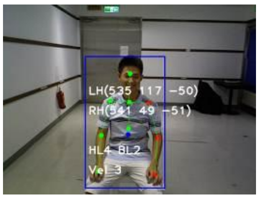
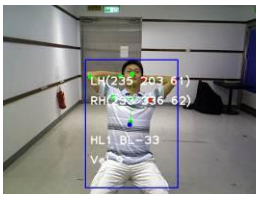
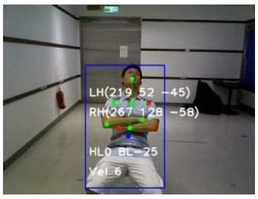
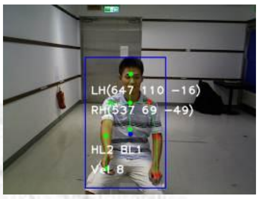
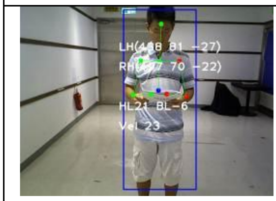
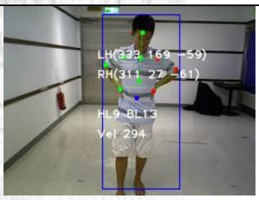
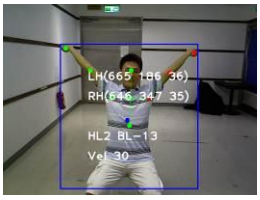
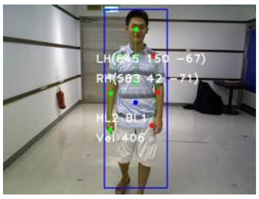
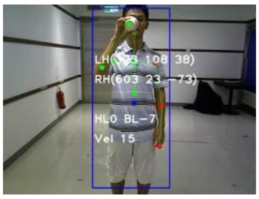
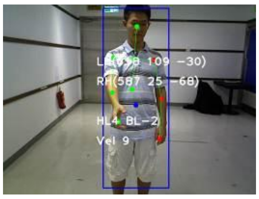
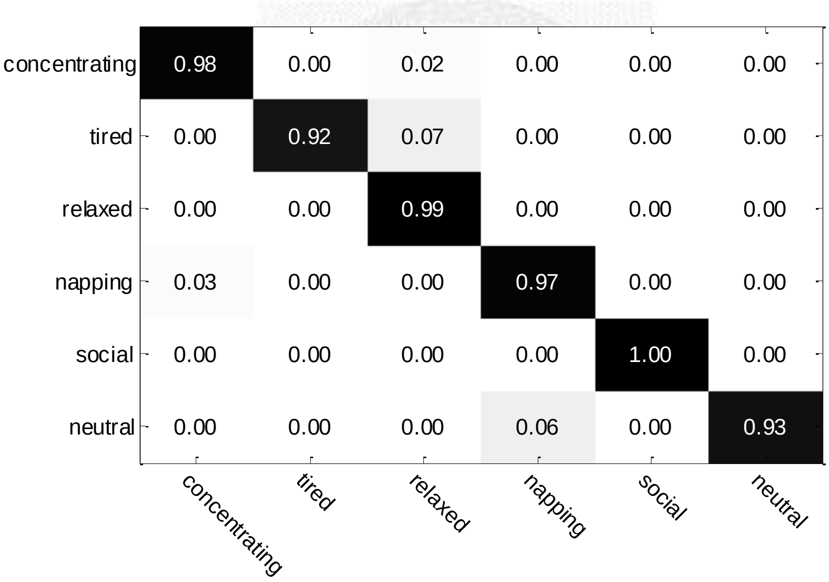
Sitting poses			
	concentrating, reading LH (509 106 -38) RH (464 79 2) HL 8 BL 10 Vel 8		tired, stretching LH (689 187 34) RH (660 348 32) HL 3 BL -18 Vel 19
	neutral, normal sitting LH (618 109 -30) RH (587 25 -68) HL 4 BL -2 Vel 9		relaxed, hands on head LH (235 203 61) RH (233 336 62) HL 1 BL -33 Vel 9
	napping, leaning backward LH (219 52 -45) RH (267 128 -58) HL 0 BL -25 Vel 6		social, hand shaking LH (647 110 -16) RH (537 69 -49) HL 2 BL 1 Vel 8
Standing poses			
	concentrating, reading LH (438 81 -27) RH (197 70 -22) HL 21 BL -6 Vel 23		tired, tired walking LH (333 169 -59) RH(311 27 -61) HL 9 BL 13 Vel 294
	tired, stretching LH (665 186 36) RH (646 347 35) HL 2 BL -13 Vel 30		neutral, normal walking LH (645 150 -27) RH (583 42 -71) HL 2 BL 1 Vel 406
	relaxed, drinking LH (323 108 38) RH (603 23 -73) HL 0 BL -7 Vel 15		social, hand shaking LH (618 109 -30) RH (587 25 -68) HL 4 BL -2 Vel 9

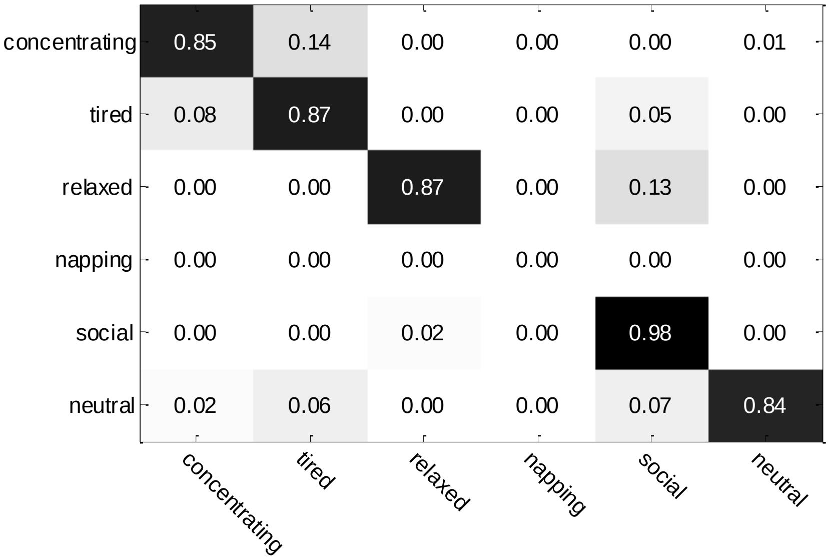
Fig. 4-11. Example views of captured user postures. In this figure, left and right are viewed from observer’s side.

To test the discriminating ability of proposed skeleton-based features, we performed the leave one person out (LOPO) testing scenario. Sitting and standing cases are tested separately because whether a user is sitting or standing can very efficiently discriminated by the height of the user's head above ground. We used SVM with RBF kernel as the classifier. The accuracy of sitting is 97.20±2.65%, and that of standing is 90.39±12.76% for untrained user. The dataset used here is not exactly the same as the one used in HOG feature, but actors were asked to perform similar actions. So, we can say that our features is a great improvement from using HOG feature in terms of invariance over different users. Besides, our features only composed of 9 values in total.



	concentrating	tired	relaxed	napping	social	neutral
concentrating	0.98	0.00	0.02	0.00	0.00	0.00
tired	0.00	0.92	0.07	0.00	0.00	0.00
relaxed	0.00	0.00	0.99	0.00	0.00	0.00
napping	0.03	0.00	0.00	0.97	0.00	0.00
social	0.00	0.00	0.00	0.00	1.00	0.00
neutral	0.00	0.00	0.00	0.06	0.00	0.93

Fig. 4-12. Confusion matrix of sitting case using skeleton-based feature, leave one person out (user independent classification).



	concentrating	tired	relaxed	napping	social	neutral
concentrating	0.85	0.14	0.00	0.00	0.00	0.01
tired	0.08	0.87	0.00	0.00	0.05	0.00
relaxed	0.00	0.00	0.87	0.00	0.13	0.00
napping	0.00	0.00	0.00	0.00	0.00	0.00
social	0.00	0.00	0.02	0.00	0.98	0.00
neutral	0.02	0.06	0.00	0.00	0.07	0.84

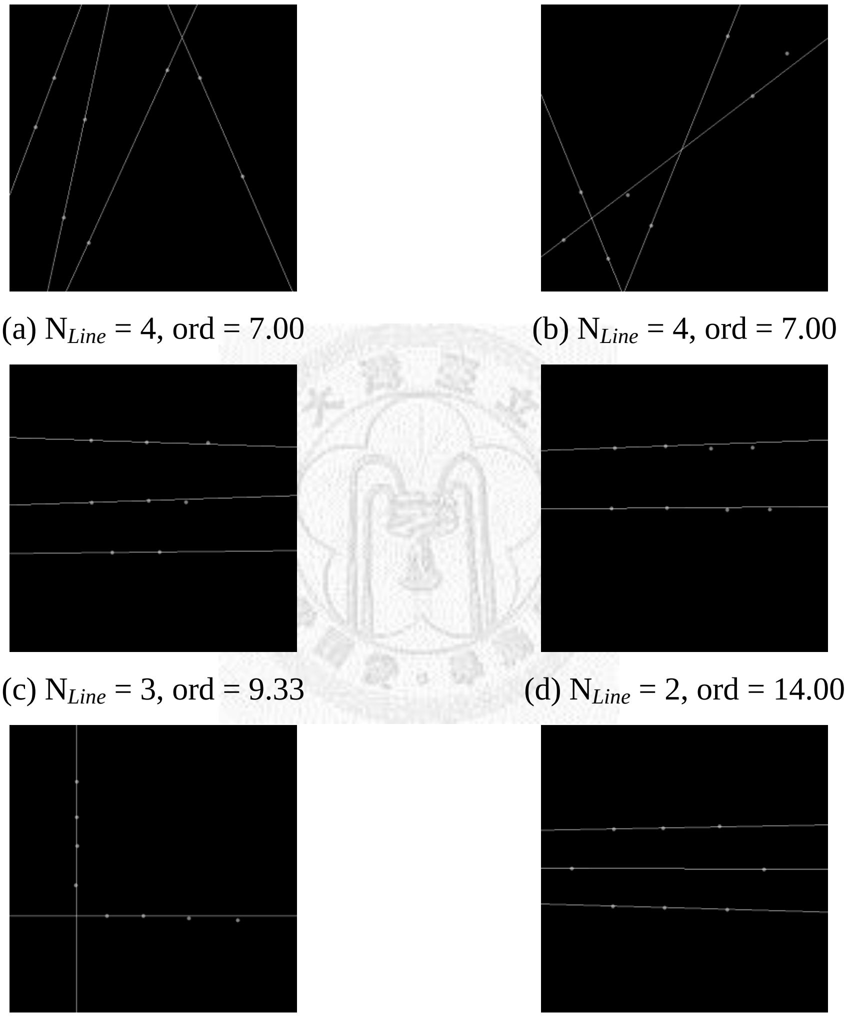
Fig. 4-13. Confusion matrix of standing case using skeleton-based feature, leave one person out (user independent classification).

4.1.3 Human-Human Interaction

4.1.3.1 Orderliness feature

Here we show six examples to test our proposed orderliness feature as shown in Fig. 4-14(a-f). In each sub-figure, 8 solid circles represent 8 people on the 2D ground with different configurations, and the lines show a solution of using minimum number of lines to fit the people. Fig. 4-14(a)(b) simulate social context, where people prone to randomly form clusters. In (b), the minimum number of lines to fit all the people (N_{Line}) is 4, in which 3 of them are explicitly shown on the image and 1 of them is degenerated to a point that fits the point at the upper-right corner. Fig. 4-14(c)(d)(e)(f) simulate formal meeting activities (classified as concentrating context), where people sits or stands quite orderly. The values of N_{Line} are 3, 2, 2 and 3 respectively. The value of orderliness features (ord) is calculated using (2.15). As shown in the figure, orderliness of (c), 9.33, (d), 14.00, (e), 14.00 and (f), 9.33, are greater than that of (a) and (b), 7.00. Therefore, this orderliness feature is effective to discriminate social or concentrating context in a scene with multiple human.

The orderliness feature provides a cue to discriminate between social and concentrating context in a multi-people scene. To evaluate our proposed orderliness feature, we captured a 3.5-hour-long RGB-D video of our lab’s seminar. The full video is segmented into 35 6-minute-long sequences, and each sequence is labeled with its context. In Fig. 4-15, human detection and orderliness feature extraction were performed on four sequences. In (a) and (b), mean orderliness values are similar, 5.98, and 5.61 respectively, and are larger than those of (c) and (d), 2.47, and 2.56 respectively. The variance of (a) is larger than (b) is probably because before seminar, some people were already seated but others were still walking around chatting, which results in a mixture of both context. On the other hand, the corresponding histogram of orderliness feature extracted from each frame also shows the tendency that the concentrating context has the distribution at larger orderliness values than social context does.



(a) N_{Line} = 4, ord = 7.00

(b) N_{Line} = 4, ord = 7.00

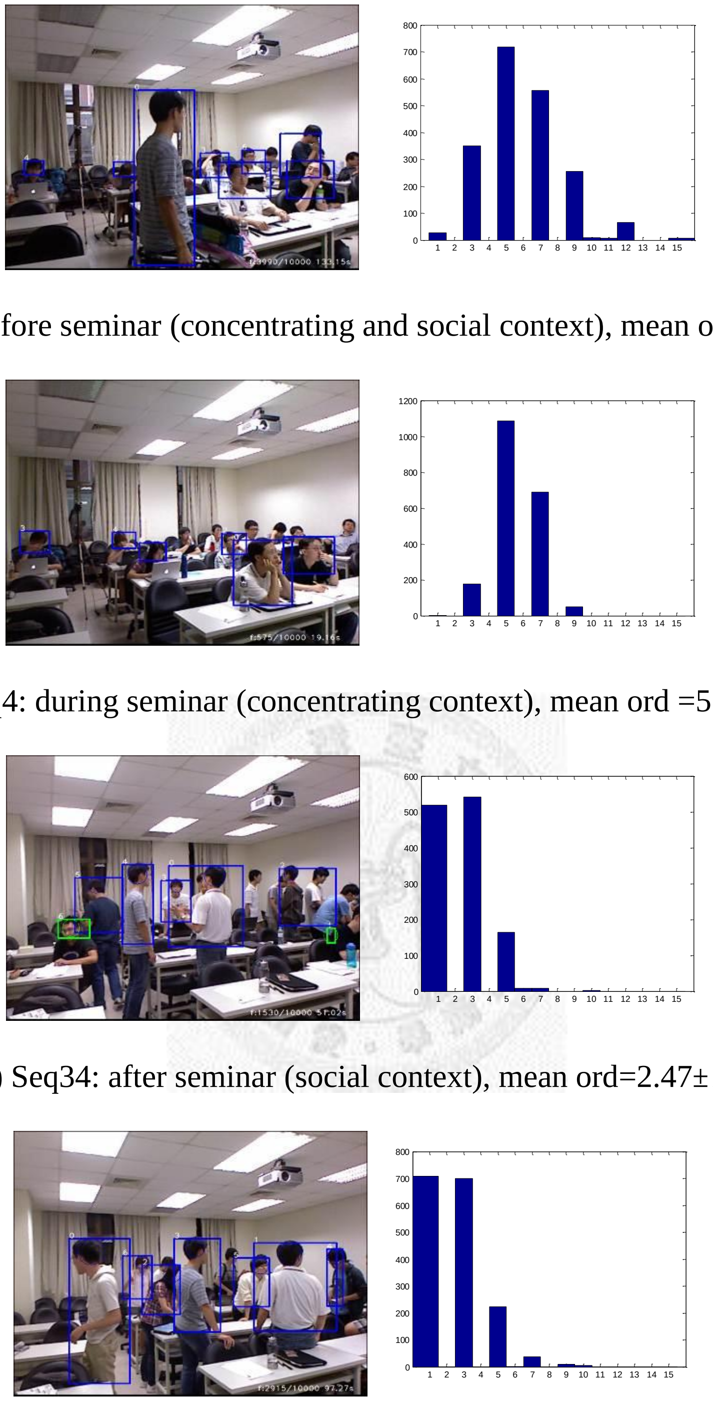
(c) N_{Line} = 3, ord = 9.33

(d) N_{Line} = 2, ord = 14.00

(e) N_{Line} = 2, ord = 14.00

(f) N_{Line} = 3, ord = 9.33

Fig. 4-14. Four simulated examples of orderliness feature extraction. Solid circles indicate people's position. Lines show a solution of using minimum number of lines to fit the people.



(a) Seq1: before seminar (concentrating and social context), mean ord=5.98±2.32

(b) Seq4: during seminar (concentrating context), mean ord =5.61±1.32

(c) Seq34: after seminar (social context), mean ord=2.47±1.53

(d) Seq35: after seminar (social context), mean ord=2.56±1.66

Fig. 4-15. Four real examples of orderliness feature extraction. *Left*: sampled image of the sequence, and people detection result. *Right*: histogram of orderliness feature extracted from frames in the sequence.

4.2 Human-Context Mapping

In this section, we evaluate the efficacy of our human-context mapping algorithm. The efficacy is defined as the rewards collected by the robot in the execution of service in a period. In order to fully demonstrate the superiority of our system, we need an office environment which is large enough (*e.g.* more than 10 people), and the mapping duration should be sufficiently long (*e.g.* more than 1 month). However, this is requires great effort to employ our robot in real environment. Therefore, we begin by building a simulation environment as the platform to tune the parameters and evaluate the result.

4.2.1 Simulation

4.2.1.1 Environment Setup

The simulated office environment is based on occupancy grid map and incorporates possible locations for humans to stay. As shown in Fig. 4-16, the dark regions indicate inaccessible places or obstacles. The occupancy map was originally built using Laser Range Finder at the second floor of Minda Building in National Taiwan University and later manually refined to form a clearer version. The blue markers show all the possible locations where human may stay. There may be a desk, sofa, or meeting tables. Below we will refer to these locations as people staying locations (PSLs). Notice that we include the direction of each location only for robot to serve people from their socially acceptable direction (*e.g.* from left or right but not from behind). These details are counted because in real cases every movement requires ineligible time. And time is a key factor of our algorithm. By doing so, we can make our simulation environment to be as closer to real environment as possible.

When the simulation begins, a robot will be sent out to explore the surrounding and actively serves people according to the context inferred. Here, we assume the context inference is given but with some noises to simulate imperfect sensing in real cases. So, each PSL will provide a noisy human-context inference result represented by $\mathbf{P}(E)$ at the current timestamp. Also, robots can only make inference from observation at places inside its camera's field of view.

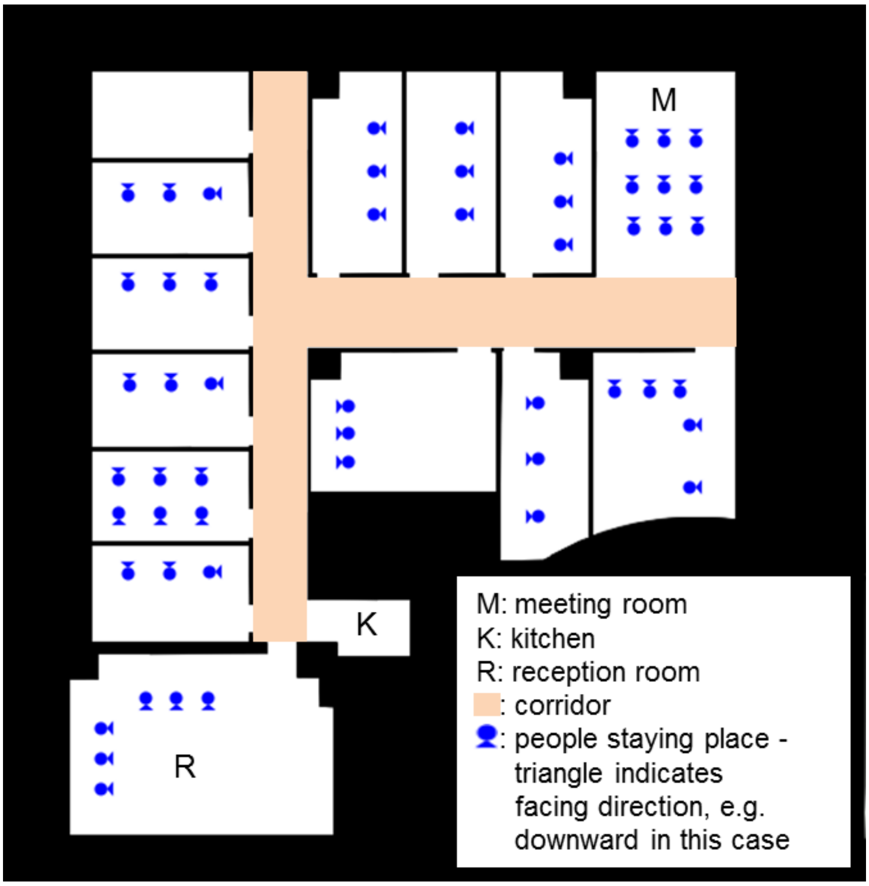
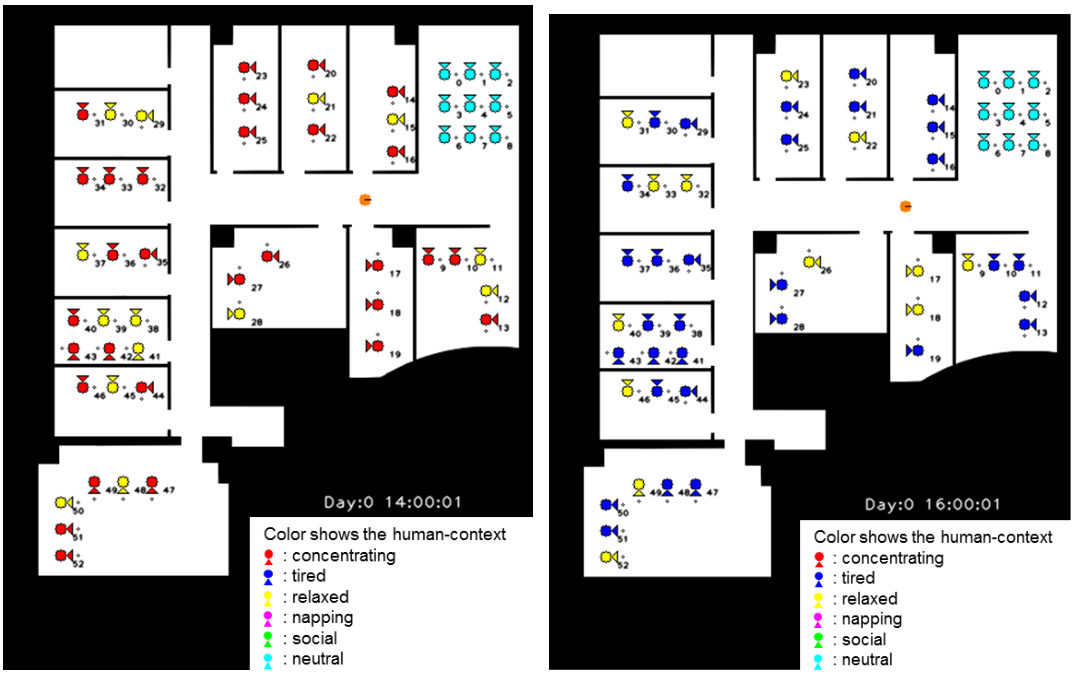
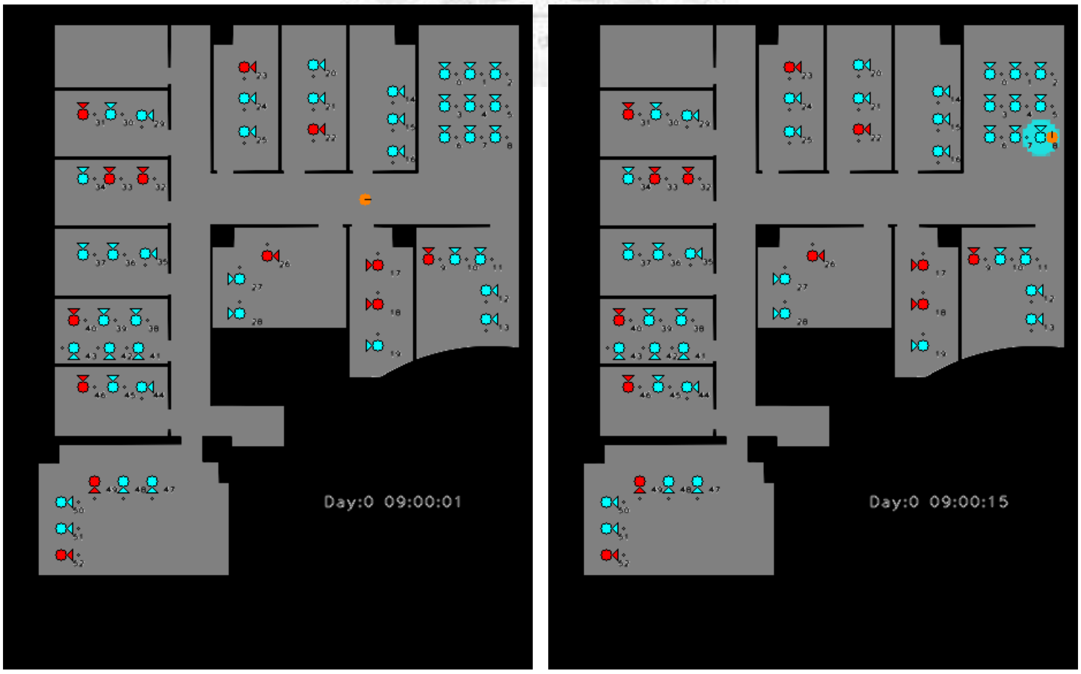

M
K
R
M: meeting room
K: kitchen
R: reception room
: corridor
: people staying place - triangle indicates facing direction, e.g. downward in this case

Fig. 4-16. The simulation environment.

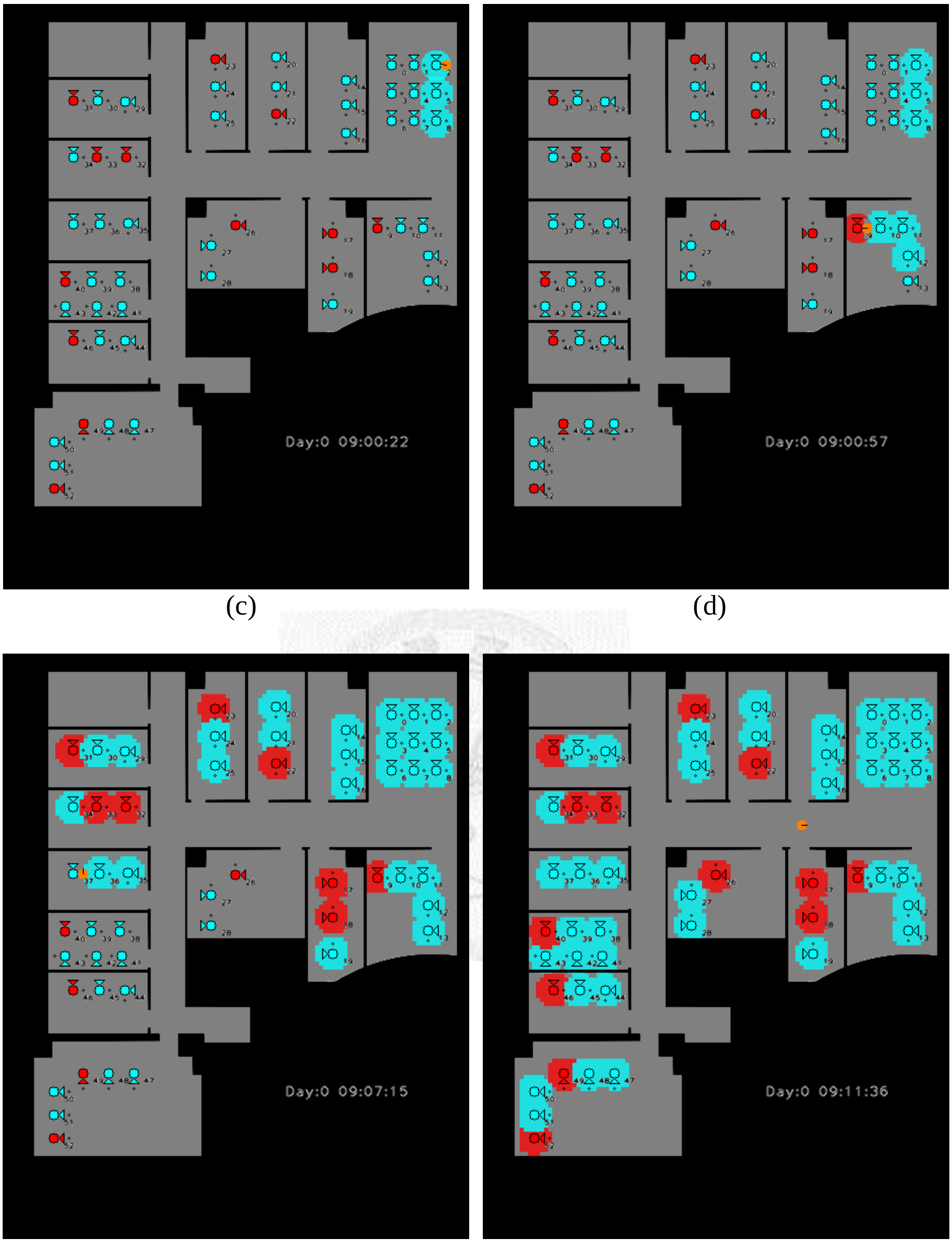


(a) (b)

Fig. 4-17. Snapshots of the simulation. Date is shown at bottom right. The number besides people is the identification number of a staying location. Color on the people shows the ground-truth of human-context.



(a) (b)



(c) (d)

(e) (f)

Fig. 4-18. Context mapping. (a) beginning of mapping, (b) robot recognized neutral context and updated the map, (c) robot exploring the meeting room, (d) robot recognized tired context and updated the map, (e)(f) following exploration. Color on map shows the Context mapping result. *Cyan*: neutral, *Blue*: tired, *Gray*: no information, *Orange*: robot location.

4.3 Context Aware Servicing

Here we integrate the human-context recognition ability, context mapping ability, and service planning to test our robot in a realistic scenario. Three staying places, A (manager's office), B (boss's office), and M (meeting room) are in an office environment as shown in Fig. 4-19. Users will follow the schedule of human context shown in Fig. 4-20 to perform corresponding pose. For fast evaluation we only simulate users' actions between 13:00 and 14:00 in work days. The schedule was performed several times to simulate several days.

The robot has two modes: one is mapping and passive servicing (MPS) mode, and the other is mapping and active servicing (MAS) mode. In MPS mode, robot performs depth-first traversal to traverse the whole environment and records human context on the Historical map. Also, it serves users on user's calling. In MAS mode, it will actively serve the users according to its inference of human context. The robot switched between the two modes by examining whether the entropy of the Instant map inferred (E) is higher than a threshold or not. If entropy is high, the robot will be in MPS mode, otherwise, in MAS mode.

At the first arrival at the new office, the robot did not have any information in its Historical map (M) defined in Section 3.2 . So in the first few days, the robot was in the MPS mode and wandering around. Fig. 4-21 shows several snapshots of the robot observing people in its first working day. (a) Robot observed the manager at A was concentrating at 13:10. (b) Robot observed the boss at B was napping at 13:15. (c) Robot observed the boss at B was tired at 13:26 in the first day. (d) Robot observed the people at M was in social context at 13:42 in the first day. After several observations, the robot had accumulated sufficient information and switched into MAS mode and provided services actively.

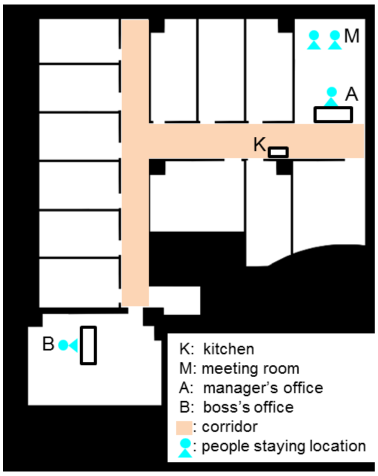
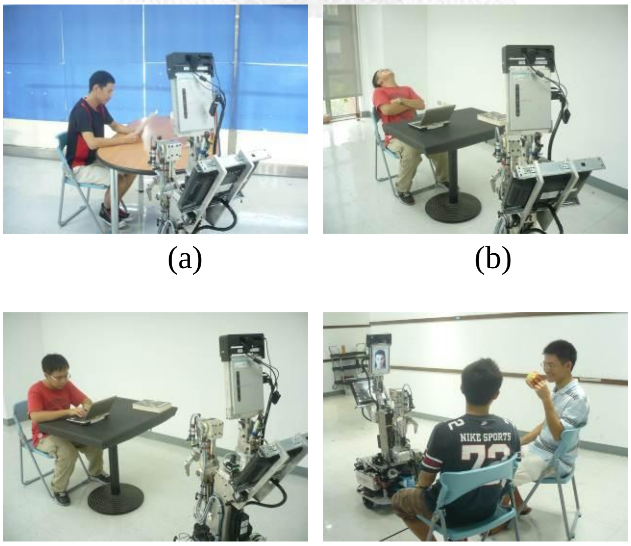


Fig. 4-19. Environment for overall test.

13:00 ***13:20*** ***13:40*** ***14:00***

A	Concentrating	Concentrating	Relaxed
B	Napping	Relaxed	Tired
M	Neutral	Neutral	Social

Fig. 4-20. Schedule of human context in testing environment.

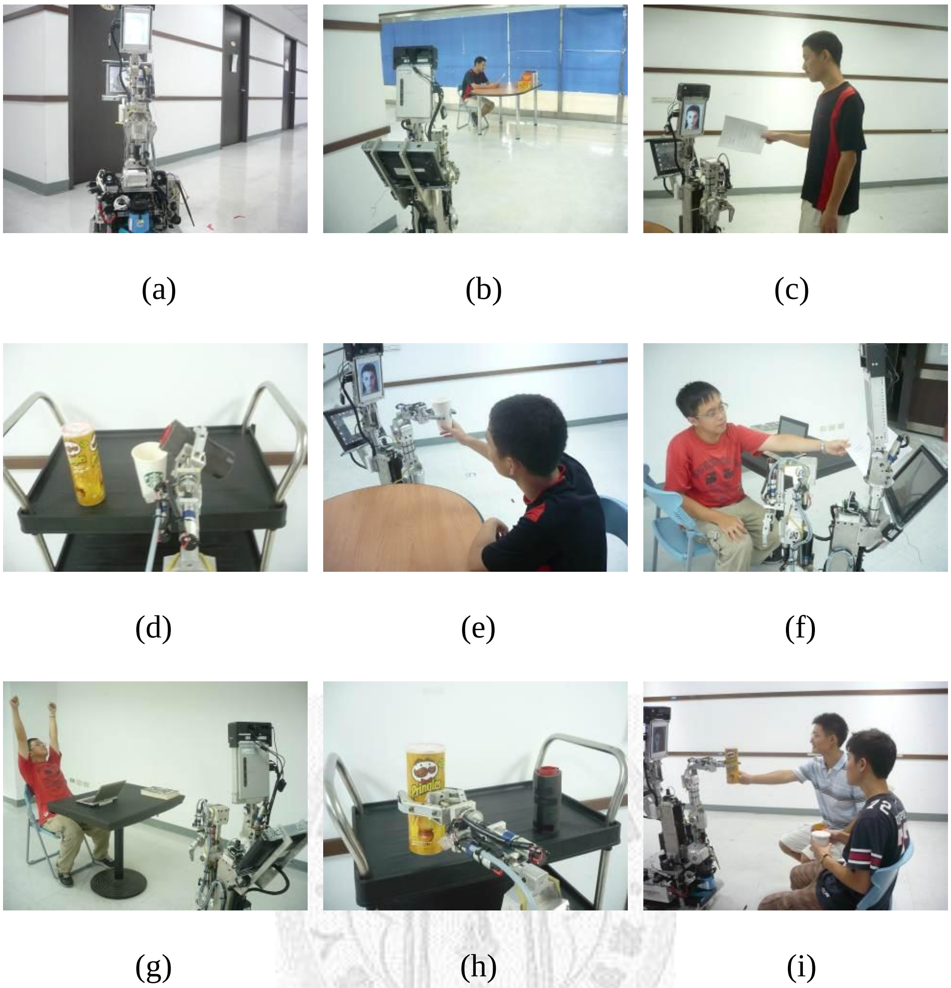


(a) (b)

(c) (d)

Fig. 4-21. Robot wandered and observed human context to build the Historical map.

After several observations in one day (actually a day with only 1 hour), robot switched into MAS mode. Fig. 4-22 shows the behaviors performed by robot, and we described it in detail as follows. In the second day, the robot started its work on the cooridor (Fig. 4-22 (a), 13:00). The manager at A called the robot via a controller (13:00). The robot approached A and received the task of delivering the document to the boss at B (Fig. 4-22 (b)(c), 13:03). However, according to robot's human context map, it inferred that the boss was currently taking a nap, so it planned to serve coffee to the manager before delivering the document to the boss. It planned so because this earns more rewards in the predicted interval than going directly toward the boss. It prepared coffee at K (13:08) After the coffee was deliverred to the manager (13:15), it planned to carry the document to the boss because according to the prediction the boss would be awake and in concentrating mode when it arrived at B. After it delivered the document (Fig. 4-22 (f), 13:21), it observed that the boss was tired from his pose (Fig. 4-22(g)), so it provided play_music and book_massage (13:23). By the rule that robot should not serve one person for too long the robot planned to serve snacks to C (13:30), because it predicted that there would be social context occuring around 13:40. Before it went to C, it planned to grab the snack at K, which eliminated a great amount of time traveling. If the robot had had no prior information from Historical map, it would have had to go to C, and made observation and then moved to K to grab the snack. The robot delivered the snack to the place C (Fig. 4-22(i), 13:45). Afterwards, it reported news for the users at C, because of the inferred social context.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4-22. Robots planned and provided services according to Human-Context Instant map and the utility function.

Chapter 5

Conclusion

We propose a context-aware robot service system that enables the robot to infer human context in office environment so as to behave and service properly. By questionnaire, we define 6 context categories for office environment, including *concentrating, tired, relaxed, napping, social,* and *neutral*. Three major contributions are listed as follows:

First, we design several novel and discriminative features for inferring human context from visual observation. The features incorporate cues from human pose, human-object interaction, and human-human interaction.

Second, to infer human context more efficiently, we propose a novel human-context mapping framework that records human context into a spatial-temporal grid map. The map represents the relationship of human context with location and time. We construct the inference and mapping framework using Dynamic Bayesian Network.

Third, robot behavior decisions are integrated into the framework. A unified dynamic decision network is applied for the robot to plan proper services according to the Human-Context Instant map.

In the experiment, using our proposed skeleton-based features can recognize human context with an accuracy of 93.75%. The overall system has been tested in a real scenario, and the result shows the correctness and efficiency of our context-aware robot service system.

REFERENCE

[1] J. Civera, D. Galvez-Lopez, L. Riazuelo, J. D. Tardos, and J. M. M. Montiel, "Towards semantic SLAM using a monocular camera," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[2] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Ruhr, M. Tenorth, and M. Beetz, "Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[3] E. R. Nascimento, G. L. Oliveira, M. F. M. Campos, and A. W. Vieira, "Improving Object Detection and Recognition for Semantic Mapping with an Extended Intensity and Shape based Descriptor," in *IROS 2011 workshop - Active Semantic Perception and Object Search in the Real World*, 2011.

[4] J. G. Rogers, A. J. B. Trevor, C. Nieto-Granda, and H. I. Christensen, "Simultaneous localization and mapping with learned object recognition and semantic data association," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[5] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, pp. 915-926, 2008.

[6] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, and S. Thrun, "Learning hierarchical object maps of non-stationary environments with mobile robots," in *Proc. of the Eighteenth conference on Uncertainty in artificial intelligence*, 2002.

[7] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, pp. 927-941, 2008.

[8] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based Object Labeling in 3D Scenes," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[9] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *Neural Information Processing Systems*, 2011.

[10] X. Ren, L. Bo, and D. Fox, "RGB-(D) Scene Labeling: Features and Algorithms," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2012.

[11] A. Rottmann, O. M. Mozos, C. Stachniss, and W. Burgard, "Semantic place classification of indoor environments with mobile robots using boosting," in *National Conference on Artificial Intelligence*, 2005.

[12] Ó. Martínez Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, pp. 391-402, 2007.

[13] H. Zender, O. Martínez Mozos, P. Jensfelt, G. J. M. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, pp. 493-502, 2008.

[14] C. Galindo, J.-A. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," *Robotics and Autonomous Systems*, vol. 56, pp. 955-966, 2008.

[15] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots—an object based approach," *Robotics and Autonomous Systems,* vol. 55, pp. 359-371, 2007.

[16] S. Vasudevan and R. Siegwart, "Bayesian space conceptualization and place classification for semantic maps in mobile robotics," *Robotics and Autonomous Systems,* vol. 56, pp. 522-537, 2008.

[17] M. Pantic, A. Pentland, A. Nijholt, and T. Huang, "Human Computing and Machine Understanding of Human Behavior: A Survey," in *Artifical Intelligence for Human Computing*. vol. 4451, T. Huang, A. Nijholt, M. Pantic, and A. Pentland, Eds., ed: Springer Berlin / Heidelberg.

[18] I. Laptev and G. Mori. (2010). *Statistical and Structural Recognition of Human Actions* [Online]. Available: https://sites.google.com/site/humanactionstutorialeccv10/

[19] Z. Zhihong, M. Pantic, G. I. Roisman, and T. S. Huang, "A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 39-58, 2009.

[20] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009.

[21] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the Viewpoint Feature Histogram," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[22] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *The International Journal of Robotics Research,* vol. 31, pp. 647-663, 2012.

[23] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *The International Journal of Robotics Research,* vol. 26, pp. 889-916, 2007.

[24] A. Pronobis, O. Martínez Mozos, B. Caputo, and P. Jensfelt, "Multi-modal Semantic Place Classification," *The International Journal of Robotics Research,* vol. 29, pp. 298-320, 2010.

[25] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert, "From 3D scene geometry to human workspace," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[26] G. Debord, *Introduction to a critique of urban geography,* 1955.

[27] C. Nold, "Emotional Cartography," *Bio Mapping website,* 2009.

[28] M. Coulson, "Attributing Emotion to Static Body Postures: Recognition Accuracy, Confusions, and Viewpoint Dependence," *Journal of Nonverbal Behavior,* vol. 28, pp. 117-139, 2004.

[29] A. Kapur, A. Kapur, N. Virji-Babul, G. Tzanetakis, and P. Driessen, "Gesture-Based Affective Computing on Motion Capture Data," in *Affective Computing and Intelligent Interaction,* 2005.

[30] N. Sebe, I. Cohen, and T. S. Huang, "Multimodal emotion recognition," *Handbook of Pattern Recognition and Computer Vision,* vol. 4, pp. 387-419, 2005.

[31] T. Baltrusaitis, D. McDuff, N. Banda, M. Mahmoud, R. el Kaliouby, P. Robinson, and R. Picard, "Real-time inference of mental states from facial expressions and upper body gestures," in *IEEE Int. Conf. on Automatic Face & Gesture*

Recognition and Workshops, 2011.

[32] B. Yao and F.-F. Li, "Recognizing Human-Object Interactions in Still Images by Modeling the Mutual Context of Objects and Human Poses," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 99, 2012.

[33] T. Lan, Y. Wang, W. Yang, S. Robinovitch, and G. Mori, "Discriminative Latent Models for Recognizing Contextual Group Activities," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. PP, pp. 1-1, 2011.

[34] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing,* vol. 28, pp. 976-990, 2010.

[35] *The 12 Dramatic Elements*. Available: http://www.thedramateacher.com/dramatic-elements/

[36] *Elements of Drama - Notes*. Available: http://www.slideshare.net/cathtallks/elements-of-drama-notes

[37] T. Hofmann, "Unsupervised Learning by Probabilistic Latent Semantic Analysis," *Machine Learning,* vol. 42, pp. 177-196, 2001.

[38] S. Bullon, "Longman dictionary of contemporary English," ed: Pearson Longman, 2009.

[39] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression (2nd ed.)*: Wiley, 2000.

[40] P. H. Wicksteed, *The Common Sense of Political Economy*, 1910.

[41] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *IEEE 12th Int. Conf. on Computer Vision*, 2009.

[42] Y. Jianchao, Y. Kai, G. Yihong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[43] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 21, pp. 433-449, 1999.

[44] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Learning informative point classes for the acquisition of object model maps," in *Int. Conf. on Control, Automation, Robotics and Vision.*, 2008.

[45] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 67, pp. 301-320, 2005.

[46] D. Bertsekas, *Nonlinear Programming*: Athena Scientific, 1999.

[47] K. Lai, B. Liefeng, R. Xiaofeng, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[48] Y. L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[49] Y. L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *ICML*, 2010.

[50] J. Yang and M. H. Yang, "Learning Hierarchical Image Representation with Sparsity, Saliency and Locality," *BMVC*, 2011.

[51] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, vol. 20, pp. 1254-1259, 1998.

[52] K.-T. Yu and L.-C. Fu, "Learning Hierarchical Representation with Sparsity for RGB-D Object Recognition," in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, to be published.

[53] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[54] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[55] C. Thurau and V. Hlavac, "Pose primitive based human action recognition in videos or still images," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[56] L. Wei-Lwun and J. J. Little, "Simultaneous Tracking and Action Recognition using the PCA-HOG Descriptor," in *The 3rd Canadian Conf. on Computer and Robot Vision*, 2006.

[57] *OpenNI*. Available: http://www.openni.org/

[58] Z. Ghahramani, "Learning dynamic Bayesian networks," 1998.

[59] J. Yang and M. H. Yang, "Learning Hierarchical Image Representation with Sparsity, Saliency and Locality," *BMVC 2011*.

[60] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871-1874, 2008.

[61] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1627-1645, 2010.