A FRAMEWORK FOR PERSONAL EMOTION CATEGORIZATION AND VISUAL ATTENTION ESTIMATION

HUANG XUELIN

Ph.D

The Hong Kong Polytechnic University

2016

The Hong Kong Polytechnic University
Department of Computing

A Framework for Personal Emotion Categorization and Visual Attention Estimation

Huang Xuelin

A thesis submitted in partial fulfillment of the
requirements
for the Degree of Doctor of Philosophy

February 2016

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

Huang Xuelin (Name of student)

Abstract

Visual signals, such as those obtained by observing facial expression and eye movements, are keys to understanding how humans think and feel. There has therefore been much previous work in facial expression analysis and eye gaze analysis, but the work is hampered by two main challenges: human behavior varies a lot, which makes it hard to generalize across multiple individuals; and data annotation is expensive, therefore it is very difficult to collect large amounts of data from which to generalize.

In my thesis work, I address these challenges through a framework for personal emotion categorization and visual attention estimation. I establish several different approaches for constructing accurate user-dependent models, which are designed to address the challenge of personal differences in facial affect and visual attention estimation problems. My work focuses on a non-intrusive approach that would be suitable for *in-situ* contexts, without the need for specialized hardware.

For facial affect recognition, to feasibly acquire adequate target data and maximally alleviate the annotation effort for learning, I propose PADMA, an efficient association-based multiple-instance learning approach for facial affect recognition with coarse-grained annotations. I then proceed to empirically demonstrate that my proposed user-dependent models considerably outperform the state-of-the-art counterparts in facial affect recognition issues across different facial datasets. I then further extend my investigations to produce *fast*-PADMA, which addresses the effectiveness of two types of user-dependent models: the user-specific model that learns only from the target user's data, and the user-adaptive model that is trained on both the target and the source subjects. Each model has its own advantages. Given sufficient personal data, the user-specific model can fully accommodate the diverse aspects of the target user, including the facial geometry as well as the expression preference. The user-adaptive model, on the other hand, is able to adapt knowledge from a large number of source subjects, and thus requires relatively little target-specific data to achieve a satisfactory performance, which accelerates the learning process. Depending on the amount of target-

specific data available for a particular context, we can select the most appropriate form of the user-dependent model. My findings, therefore, suggest that it is feasible to build a well-performing user-dependent facial affect model for a particular user with only a limited amount of coarse-grained annotations.

For visual attention, I will use experiments to illustrate the correlation between eye gaze behavior and interactions in daily human-computer activities, such as mouse-click and keypress, which show that these correlations are dependent on the context as well as on user affect. I will then further demonstrate through PACE, which refines and adopts daily interaction-informed data for gaze learning in an implicit manner, without the need of user annotation nor intrusive calibration. Likewise, the coordination pattern between gaze movement and mouse-click is also indicative of the mental states, such as stress. The results show success in learning the visual attention location from the noisy interaction-informed data, and suggest promise in using gaze and click coordination pattern to infer stress level.

List of Publications

Michael Xuelin Huang, Grace Ngai, Kien A. Hua, Stephen C.F. Chan, Hong Va Leong. 2015. "Identifying User-Specific Facial Affects from Spontaneous Expressions with Minimal Annotation". To appear in *IEEE Transactions on Affective Computing*.

Michael Xuelin Huang, Jiajia Li, Grace Ngai, Hong Va Leong. 2016. "StressClick: Sensing Stress from Gaze-Click Patterns". To appear in *Proceedings of ACM International Conference on Multimedia*. ACM Press.

Michael Xuelin Huang, Tiffany C.K. Kwok, Grace Ngai, Stephen C.F. Chan, Hong Va Leong. 2016. "Building a Personalized, Automatically Calibrating Eye Tracker from User Interactions". In *Proceedings of the ACM annual conference on Human Factors in Computing Systems*. ACM Press 5169-5179. – **Best Paper Award**.

Michael Xuelin Huang, Tiffany C.K. Kwok, Grace Ngai, Hong Va Leong, Stephen C.F. Chan. 2014. "Building a Self-Learning Eye Gaze Model from User Interaction Data". In *Proceedings of ACM International Conference on Multimedia*. ACM Press, 1017-1020.

Michael Xuelin Huang, Will W. W. Tang, Kenneth W. K. Lo, C. K. Lau, Grace Ngai, and Stephen C.F. Chan. 2012. "MelodicBrush: a novel system for cross-modal digital art creation linking calligraphy and music." In *Proceedings of the Designing Interactive Systems Conference*. ACM Press, 418-427.

Michael Xuelin Huang, Will Tang, Kenneth W.K. Lo, C.K. Lau, Grace Ngai, and Stephen C.F. Chan. 2012. "MelodicBrush: a cross-modal link between ancient and digital art forms." In *Proceedings of ACM annual conference extended abstracts on Human Factors in Computing Systems*. ACM Press, 995-998.

Tiffany C.K. Kwok, **Michael Xuelin Huang**, Wai Cheong Tam and Grace Ngai. 2015. "Emotar: Communicating Feelings through Video Sharing". In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM Press, 374-378.

Hugo Jiawei Sun, **Michael Xuelin Huang**, Grace Ngai, Stephen C.F. Chan. 2014. “Nonintrusive Multimodal Attention Detection”**.** In *Proceedings of the 7th International Conference on Advances in Computer-Human Interactions*.

Yujun Fu, Hong Va Leong, Grace Ngai, **Michael Xuelin Huang**, Stephen C.F. Chan. 2015. “Physiological mouse: Towards an emotion-aware mouse”. To appear in *Universal Access in the Information Society*, Springer.

Yujun Fu, Hong Va Leong, Grace Ngai, **Michael Xuelin Huang**, Stephen C.F. Chan. 2014. “Physiological Mouse: Towards an Emotion-Aware Mouse”. In *International Workshop on User Centered Design and Adaptive Systems*. IEEE.

Kenneth W.K. Lo, Chi Kin Lau, **Michael Xuelin Huang**, Wai Wa Tang, Grace Ngai, Stephen C.F. Chan. 2013. "Mobile DJ: a Tangible, Mobile Platform for Active and Collaborative Music Listening". *International Conference on New Interfaces for Musical Expression*.

Acknowledgments

I would like to express my sincere gratitude to all the people that have assisted me to complete this degree. The following acknowledgments are by no means exhaustive, for which I apologize.

I would love to thank Dr. Grace Ngai for being the best advisor I could have wished for. She gave me the wonderful experience to be in the CHILab for these years with unwavering support and encouragement.

I would also like to thank the professors in my research group: Dr. Hong-Va Leong, Dr. Alvin Chan, and especially my co-supervisor Dr. Stephen C.F. Chan. They generously shared the knowledge, experience, and inspiring thoughts in our every week discussion. This work would never have been done without their generous contributions.

I am also deeply grateful to Prof. Kien A. Hua for being a fantastic supervisor and friend, during the period of my visit to the University of Central Florida.

I have had great pleasure working with members in CHILab: Dr. Yuanyuan Wang, Will Tang, Kenneth Lo, Kin Lau, Tiffany Kwok, Eugene Fu, Andy Tam, Hugo Sun and Georgia Li. The creativity of all my colleagues has been a constant inspiration throughout my time.

I also want to thank the members of my examination committee, Dr. Hatice Gunes from the University of Cambridge, Prof. Helen Meng from the Chinese University of Hong Kong, and our department head Prof. Jian-nong Cao, for their constructive comments and suggestion.

My utmost thanks go to my parents and my wife, who unconditionally support me in all my decisions.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1997, Rosalind Picard's book "Affective Computing" [85] attracted much attention. The idea of computers that could understand emotions and simulate empathy stoked the imaginations of people. Ever since then, much research work has endeavored to address the relevant issues of teaching computers to interpret human affect, which refers to the experience of emotion and is also sometimes used to refer to affect displays such as emotional facial and vocal expressions. Since affect is a fundamental component of human expression and communication [121], understanding human affect is a prominent topic in human-computer interaction and affective computing. Given the techniques of enabling machines to be aware of the users' mental states, diverse opportunities, such as friendlier and mentally healthier interface, can then be realized to facilitate advanced interaction.

Although certain affect detection techniques show promising results in exploring mental states, not all of them are suitable for daily interaction scenarios, due to various limitations, including cost, intrusiveness, and environmental noise. For example, to explore the human inner mental state, psychophysiologists advocate the importance of understanding the Autonomic Nervous System, which controls the unconscious act and regulates the body physiology. However, the accurate detection of physiological signal normally requires different kinds of sensors, such as the electrodes for brainwave and heartbeat measure, to be attached to the human body. In addition, the relation between physiological signal and affect can be influenced by multiple variables, such as initial physiological value, arousal degree, stimulus specificity and individual specificity [4]. The intrusiveness of physiological signals detection [9] and the heavy person-dependency, therefore, make it less appropriate for daily human-computer interaction. Some prior studies also propose to sense human state from equipped daily objects. For example, sitting posture can be recognized by a pressure-aware cushion [63]; and heartbeat can be estimated by the physiologically -aware mouse [32]. One limitation of such approaches is their reliance on special sensing devices, which are not widely used. Vocal tone [121] and speech text [50] have also been shown to be informative for emotional expression, especially in face-to-face conversations. However, there are some contexts in which these signals are not available or reliable, such as contexts in which speaking is not appropriate and environments with much ambient noise. In these situations, speech analysis may not be a proper or available choice for comprehensive

computer affect sensing. This thesis hence focuses on the non-intrusive visual cues of affect displays, in particular, the facial expression and eye gaze.

With the growing pervasiveness of vision system in the normal computer setting, observing facial expression and gaze pattern can be convenient, fully non-intrusive, and with low cost. A broad ranged research efforts have claimed that facial expression can be highly informative for interpreting human affect [121]. Numerous studies have been conducted to understand facial expression and infer arousal (activation of affect), valence (positive or negative), basic emotions like happiness, sadness, anger, fear, surprise and disgust, high-level mental states, including agreement, interest, rapport [121][9][97], personal traits, such as extraversion and agreeableness [108][10], and group-level affect [26][79]. Gaze, on the other hand, reflects human attention and cognition. Gaze movement has been pointed out to be valuable to investigate human interest [39] and engagement [81]. And there is mounting potential of gaze-aware systems in daily human-computer interaction and social interaction [39]. But do the current affect recognition techniques meet the demands of the real-use applications?

1.1 Background and Motivation

1.1.1 Understanding facial expression

In spite of the encouraging successes of previous work in affective computing [97], Various constraints have limited the widespread application of the affect detection techniques, including the model generalizability and data annotation requirement.

Take the facial affect recognition as an example, applying the existing techniques in real-use situations can be problematic because of the natural differences among individual users, especially for spontaneous expressions. Much research in this area focuses on training a user-independent facial affect model that fits the majority of users. The conventional approach relies on the supervised machine learning [121], which requires a "gold-standard" dataset annotated by human experts [60]. The assumption is that when the training dataset is large enough, the machine learning algorithm can recognize and discriminate different facial expressions across users. However, individual differences in facial appearance, ethnicity, culture, personality, and preference all affect the performance of the user-independent model (i.e. a generic classifier [94][18]). Furthermore, the identical facial expression might indicate

dissimilar affects for distinct persons. Evidence shows that applying an affect model trained on one dataset to another dataset results in a significant performance drop [121][65][78].

In contrast, given sufficient individual data, a user-dependent model should be able to achieve ideal recognition accuracy, for it can be well customized for a particular person, in term of the facial geometry and personal expression preferences. However, the vast majority of the recent studies are dedicated to improving the features and classification algorithms of the generic classifier, and scarce attention is paid to address the individual bias in a practical fashion.

Apart from the model generalizability, previous successes that focus on posed, or simulated expressions may not be extendable to spontaneous or natural expressions. Many early approaches model and evaluate affects based on these posed expressions in which the subjects, usually professional actors and actresses following specific instructions, are recorded using a near frontal view [121]. A widely used example is the Extended Cohn-Kanade dataset (CK+) [71]. However, there are significant differences between posed and spontaneous, naturally experienced expressions, as has been reported in previous work [41][66][7]. Other real-use issues such as out-of-plane head rotation [107] and illumination variations also make the recognition of affects from spontaneous expressions more challenging.

Although advanced machine learning techniques, given the sufficient amount of well-annotated data, may be able to discriminate the subtle differences of the spontaneous expressions, the annotation requirement is also a critical issue. Supervised learning methods require the manual annotation, which is expensive, tedious, error-prone, sometimes expertise-required.

Generally speaking, the needed degree of expertise depends on the annotation detail level. There are two main facial behavior description methods: the message judgment and the sign judgment [20]. Message judgment interprets the holistic expression. The corresponding annotation is therefore the implication of the whole expression. The sign judgment, on the other hand, focuses on the objective description of the components that convey emotions. A well-known application of the sign judgment is the Facial Action Coding System (FACS) [27], which describes the partial facial muscle movements by the activation level of particular facial Action Units (AUs) [27]. Due to the conflict between the annotation fineness and subjectivity, the correct annotation of AUs usually requires the agreement of multiple professional well-trained AU coders

[121][20]. This means that despite much previous work in AU-oriented studies [121][9][97], building a user-dependent model with sign judgment annotation is not feasible in the real-use situation.

We, therefore, see two major challenges to the application of facial affect identification in real use. The first is accommodating user differences, especially for spontaneous expressions. The second is collecting and annotating enough user-specific data and modeling the facial affect for a particular user in a practical manner.

Related research efforts approach these problems from different directions. Active learning selects a relatively small portion of discriminant samples for annotation so as to reduce human annotation effort [122]. Sharing a similar spirit, instead of labeling the facial implication of every video frame (snapshot), some studies suggest a few essential frames, such as those corresponding to the apex (local maximal change of facial movement), be manually labeled and the rest of the affect sequence extrapolated according to the frame similarity [60][129]. However, human expertise is still the prerequisite to identify the needed key frames. Other notable endeavors include the multiple-instance learning (MIL) that alleviates the annotation difficulty, and transfer learning that relies on model adaptation to a target user.

Compared to the conventional supervised learning that models facial affect on the frame-level labeled data, MIL learns from data with the coarse-grained segment-level annotation. A segment here refers to a video segment of facial response towards a certain stimulus which lasts for e.g. around 1-2 minutes, and each video frame in a segment is an instance. MIL techniques, therefore, allow the facial affect learning from the segment-level self-reported affect, which significantly reduces the annotation requirement. No expertise is needed to identify the key frames nor a particular facial action. The simplicity of target user's data acquisition ensures the feasibility and the practicality of a user-dependent model.

In the most prevalent MIL assumption, a video segment is positive if it contains at least one positive instance [30]. This is valid for many binary classification problems; however, when it comes to human emotions, it is not uncommon to have complex mixed feelings to occur within a segment of a couple minutes [35]. Some common affects, such as "neutral", may also occur frequently in a segment that is labeled as something other than neutral. Therefore, an effective technique to explore the segment annotation and an insight of the frame distribution inside segments are essential.

Transfer learning is another strategy made to improve the model generalizability. The conventional affect model that relies on one generic classifier learnt on the source subjects' data in the training set also has challenges with individual differences and adaptation for a particular user. The subjects available in the training set are referred to the source subjects, while the potential user/ test subject is the target user. Transfer learning assumes that some amount of the data from the target user is available, and uses it for the personal adaptation. Transfer learning assumes user-specificity [18][13], i.e. that the distribution or characteristics of the expressions and/or affects in the source dataset reflect those from the target set. But this assumption may not be valid due to the degree and diversity of the individual differences.

Another drawback of transfer learning comes from its computational cost. With few exceptions, transfer learning techniques adopt the instance-transfer approach, which re-weighs and heightens the instances similar to those of the target. However, similarity calculation and distribution estimation are computationally expensive, and the completion of these processes demands all the source instances be deployed to the target side. To mitigate the computation burden and address the data release problem, research endeavor has been devoted to personalizing the model by regression of multiple individual models of the source subjects [94]. However, the approximation of data distribution still relies on computationally expensive optimization.

With the assumption that the target user is similar to one or some of the source subjects, some transfer learning techniques generate the target model by aggregating the individual models [94], each of which is, however, more likely to suffer from overfitting than the generic counterpart, due to the limited amount of data from each given individual. Another common problem in most of the current publicly-available facial expression datasets is the lack of sufficient longitudinal individual data. This is because collecting short-term data from a good number of subjects can be relatively easy and practical, through the use of crowdsourcing platforms or laboratory experiments, but it is not easy to get data from the same individual over multiple sessions and over time. Therefore, it is also not easy to learn behaviors that are specific to individual users, as most datasets contain only limited data from each individual user. This is despite the fact that computing devices are becoming more personalized, and it is not impractical to imagine a computing device which understands patterns that are specific to its individual user. In order to enable the affect-aware application in real-use

situation, there is pressing need to investigate the practical modeling techniques for the user-dependent affect modeling.

1.1.2 Understanding eye gaze behaviors

Among the informative cues hidden in the eye gaze information, the location of visual attention (i.e. the gaze focus point in the screen coordinate) and the interaction related gaze movement pattern are especially related to the human affect in human-computer interaction.

Visual attention has been used to detect basic emotions [101] and high-level mental states, like attention and engagement [88][90]. However, the precise estimation of the gaze point usually relies on cornea reflection features, which is only accessible by the special infrared equipment. The dependency of such expensive and fatigue-inducing equipment precludes the consumer use of gaze-aware applications. To narrow the gap between gaze analysis and real-use applications, we believe that research needs to pay attention to the appearance-based gaze estimation using the off-the-shelf webcam system. Such approaches rely on the machine learning to identify the complex mapping between eye appearances and the gaze point locations, therefore, a well-performing gaze model generally requires a myriad of training data.

With few exceptions, most gaze estimation methods require calibration and non-periodical re-calibration in order to acquire sufficient learning data to accommodate lighting and head pose variances [125]. Such processes for appearance-based methods are obviously cumbersome and infeasible for real use [103].

Similar to the facial affect recognition, a user-specific gaze model has a better chance than a generic model to achieve accurate estimation given sufficient individual data for training. Fortunately, in contrast to the explicit annotation requirement of facial affect learning, the gaze annotation data can be acquired implicitly. Since there is likely a strong correlation between eye gaze and interaction cues, such as cursor and caret locations, it makes sense that the mapping between gaze features and the gaze point can be collected unobtrusively from normal computer interactions and used to recalibrate or retrain gaze estimation models. While a number of studies have demonstrated a correlation between gaze and cursor [43][64], there have been few efforts in using noisy daily interaction data for webcam gaze learning. One notable exception is that of Sugano et al. [103], which collects mouse clicks for incremental gaze learning.

In addition to the visual attention, eye gaze behaviors can offer further information strongly associated with mental stress. Intuitively, it makes sense that mental stress will affect the Central Nervous System (which controls gaze behavior) and the Somatic Nervous System (which controls cursor behavior), and affect the coordination between these two nervous systems. Consider, for example, people in stress are likely to be more anxious with their clicks, with the result that the visual attention leaves the target before the click occurs. Conversely, this pattern can also be used to detect stress.

Studies of the gaze-hand coordination in human-computer interaction have been carried out previously, however, the vast majority of the pertinent research intends to only explore the temporal and spatial consistency between gaze and cursor during web searching [14][89][37]. There are conflicting views on this consistency. Some scholars believe the cursor trajectory is a good approximation of gaze, while others point out that variation of gaze and cursor alignment is substantial. This controversy corroborates our concern that the impact of the mental state, which has largely been ignored in the previous studies, may not be negligible for the gaze-hand consistency. Therefore, we propose to explore the gaze-cursor pattern variations under different mental stress levels. We foresee this pattern can be used to infer stress as well.

1.2 Study Overview

The flow of this thesis is presented in Figure 1-1. The essential issue is to build a well-performing model for the target user. As mentioned in the previous sections, there are various constraints in the real-use situation, such as model generalizability and the annotation difficulty. This thesis, therefore, makes attempts to tackle these problems in four aspects: (1) building a user-specific model from only the target data; (2) building a user-adaptive model using both the available target data and the existing source data; (3) exploiting the implicit data acquisition and annotation mechanism; and (4) utilizing the cross-modal, user-independent features.

1.2.1 Building the user-specific model

We first propose the feasible techniques to build the user-specific model from only the data of the target user. This can be an ideal method to solve the identity bias, since the resulting model is fully customized for the characteristics of the target user.

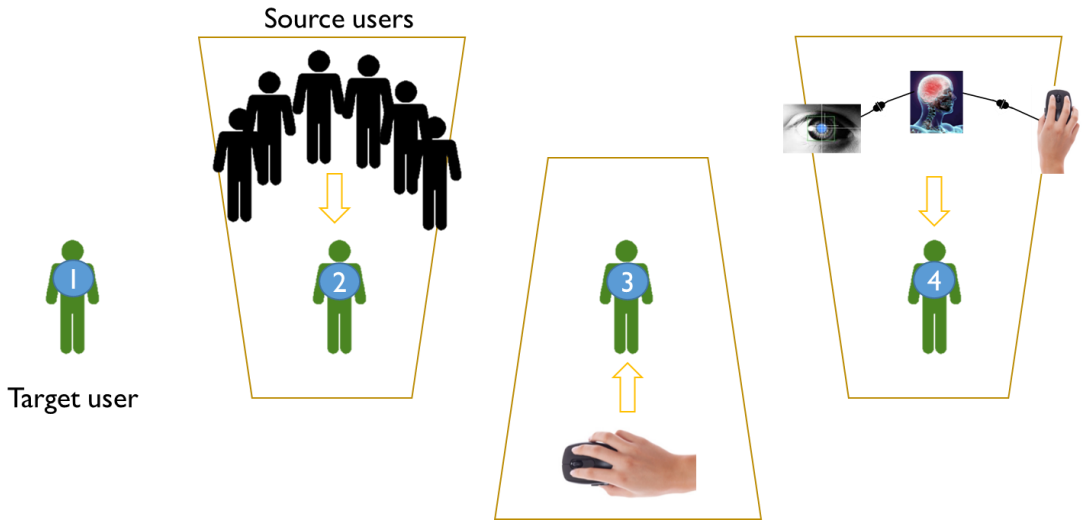


Figure 1-1. The flow of this thesis. This thesis seeks four approaches to facilitate the learning of the personalized model for a target user: (1) building a user-specific model from only the target data; (2) building a user-adaptive model using both the available target data and the existing source data; (3) exploiting the implicit data acquisition and annotation mechanism; and (4) utilizing the cross-modal, user-independent features.

We propose an approach for Personal Affect Detection with Minimal Annotation (PADMA) that uses a novel association-based multiple-instance learning (AMIL) approach. In contrast to conventional MIL methods, AMIL assumes that if an instance occurs frequently in segment(s) labeled with one particular class, but not in others, the instance has a strong association with that label.

PADMA relies on facial features similar to AUs. Similar expressions are clustered and facial gestures (defined as a short sequence of facial behavior over multiple consecutive frames [59]) extracted. AMIL is then used to correlate facial gestures with user-reported affects to obtain the fine-grained affect labels based on the distribution of the facial gestures. PADMA therefore adaptively extracts and annotates facial gestures for a user, according to his/her actual response.

Our challenge comes from identifying detailed facial gestures and their implications, given only rough overall self-reported information. The proposed method is evaluated on two spontaneous facial datasets: the publicly available UNBC-McMaster Shoulder Pain Expression Archive Database (UNBC) [72] and the Mobile Spontaneous Affect Response Video (MSARV) dataset [46], which is collected on our own for the evaluation purpose of this thesis. The results demonstrate the feasibility, effectiveness, and extensibility of our approach.

1.2.2 Building the user-adaptive model

The user-specific model learns from only the target data, therefore the acquisition of such data may be slow. We propose the user-adaptive model, exploring the possibility of using the generic knowledge extracted from the source data (from the subjects rather than the target user). The idea of this study is to utilize the similarity between the target user and the groups of source subjects who behave similarly to the target user. It is supposed that the user-adaptive model is able to achieve satisfactory performance with limited target data and therefore further lower the difficulty of building the personalized model and accelerate the learning process.

To bridge the gap between the limited individual data and a practical, well-performing affect model, this thesis proposes a novel and efficient approach to adopting the knowledge from the source data to enhance the target model. To avoid potential overfitting issues encountered in previous transfer learning methods [94], we formulate a fundamentally distinct hypothesis. We assume that there are *some* source subjects who look and behave differently from the target user and weaken the contribution of these subjects' training data to improve the source classifier. On the basis of this assumption, a radical knowledge adoption strategy is presented.

Association and statistics information are extracted for segment-level features representation. A new data alignment technique that considers the individual's expressionless, "neutral" state is used to reduce the personal geometric bias. We employ a bootstrapping technique to prepare a set of weak generic classifiers, each of which is trained on a subset of source data excluding one different subject. These weak generic classifiers are then aggregated to obtain the final recognition result. In contrast to previous studies, the proposed method, without storing the training instances nor depending on computational optimization, can effectively adapt to user individuality.

Our method has been rigorously evaluated to assess the level of pain, arousal, and valence on four publicly available datasets: UNBC-McMaster Shoulder Pain Expression Archive Database, Denver Intensity of Spontaneous Facial Actions (DISFA) [76], Mobile Spontaneous Affect Response Video and MAHNOB-HCI emotion recognition dataset (MAHNOB) [101]. The experimental evaluation demonstrates the simplicity and effectiveness of our method in aligning data of different individuals and transferring the generic knowledge.

1.2.3 Exploiting implicit data acquisition and annotation

Exploring the common patterns between the target user and source subjects is beneficial, but there are still risks to transfer knowledge across different individuals. We also seek user-specific data acquisition in an implicit and continuous manner. In doing so, sufficient data can be collected for building the personalized model without user efforts.

We take the gaze estimation as an example to study this research issue. Most previous work [103][28][48] makes the assumption that users are looking at where they click, or, to put it more broadly, that users are looking at the interaction cue (cursor or caret) at the moment that the interaction is triggered. However, this assumption may be not valid in real-use situations, due to diverse factors such as eye blink, mind-absence, response delay, individuality and task difference.

We, therefore, propose to apply behavior-informed and data-driven approaches to identify reliable training instances from daily-use interaction data and webcam video. To the best of our knowledge, there is no prior work that seeks to automatically identify and validate noisy interaction and webcam video data for gaze model learning. Given sufficient interaction data, a user-specific gaze estimation model can be built. Sharing the similar spirit of the user-specific facial affect model, the proposed gaze model can fully adapt to a particular user. This conduces to the accurate estimation of gaze point in the practical use with the off-the-shelf device.

An *in-situ* study using real-life tasks on a diverse set of interactive applications demonstrates that our Personalized, Auto-Calibrating Eye Tracker (PACE) performs comparably to state-of-the-art, but without the need for explicit training or calibration. This demonstrates the effectiveness of both the gaze estimation method and the corresponding data collection mechanism.

1.2.4 Exploiting the user-independent behaviors

Apart from increasing the amount of data for learning, using informative and robust features is also critical for a well-performing model. Although it is generally true that the user differences somehow influence the human affect displays, there are still some behaviors and responses relatively universal across people. To study the affect detection from user-independent behaviors, we investigate the cross-modal features between gaze and mouse-click for mental stress sensing.

Stress-sensing is valuable in many applications, including online learning crowdsourcing and other daily human-computer interactions. Traditional affective computing techniques investigate affect detection based on different individual modalities, such as facial expression, vocal tones, and physiological signals or the aggregation of signals of these independent modalities, without explicitly exploiting their inter-connections. In contrast, this study focuses on exploring the impact of mental stress on the coordination between two human nervous systems, the somatic and autonomic nervous systems. Specifically, we present the analysis of the subtle but indicative pattern of human gaze behaviors surrounding a mouse-click event, i.e. the *gaze-click pattern*.

Our evaluation shows that mental stress affects the gaze-click pattern, and this influence has largely been ignored in previous work. This study, therefore, further proposes a non-intrusive approach to inferring human stress level based on the gaze-click patterns, using only data collected from the common computer webcam and mouse. We conduct a human study on solving math questions under different stress levels to explore the validity of stress recognition based on this coordination pattern. Experimental results show the effectiveness of our technique and the generalizability of the proposed features for user-independent modeling. Our results suggest that it may be possible to detect stress non-intrusively in the wild, without the need for specialized equipment.

1.3 Thesis Aims and Outline

The aims of this thesis, as outlined in the overview, are as follows:

- To design the user-specific facial affect model, which requires minimal self-reported annotation, but can be fully customized for the target user.
- To propose the user-adaptive facial affect model that is learned simultaneously on both target and source subjects' data, and which is able to achieve the balance between generic knowledge and the specific adaptation in a practical manner.
- To refine the noisy daily interaction-informed data for gaze learning, in order to accurately estimate the visual attention with the off-the-shelf device.
- To identify the mental stress from variations of gaze-click pattern, using the daily interaction data to further understand the human affect.

The reminders of this thesis will cover the following material:

Chapter 2 provides the literature reviews on the facial affect recognition and the gaze analysis research work. More specifically, related research efforts to address the practical limitations are given to motivate the need for the practical personalized facial affect and gaze models.

Chapter 3 presents the user-specific facial affect modeling. This study is to demonstrate that the proposed MIL technique is useful to lower the user effort of data annotation. It therefore facilitates the learning of user-specific model in a feasible manner. Experimental results on different datasets show the effectiveness of the facial affect learning with only minimal user annotation.

Chapter 4 describes the user-adaptive facial affect learning. The motivation of the user-adaptive model is to shorten the procedure of collecting sufficient data for the personalized model. The user-adaptive model is built on both the target and source subjects' data. Evaluation results show that our technique succeeds in using the source data to improve the model performance of the target user.

Chapter 5 depicts the techniques to extract gaze features from the off-the-shelf webcam and use in conjunction with the normal, daily interaction data such as mouse-click and keypress to learn the location of visual attention. This technique demonstrates that daily interactions can be used for implicit data acquisition and annotation for human state learning, without the explicit help from the target user.

Chapter 6 presents the use of cross-modal features for mental stress detection. The coordination between gaze and mouse-click behaviors are investigated. This study shows that the cross-modal features using the human coordination between different nervous systems can be universal across people for stress detection. It therefore indicates another potential approach to enhance the personalized model.

Chapter 7 summarizes the contributions and limitations of this thesis and the potential future work. This chapter also includes a brief introduction of other contributions I have made that are beyond the scope of this thesis.

Chapter 2

Literature Review

This chapter begins with a review of the literature on the general machine learning techniques that facilitate the facial affect recognition. The studies that contribute to learning a user-dependent model will then be presented. This is followed by the review of gaze related studies, including the generalization and the implicit collection of the target user's gaze data and the work on the gaze-cursor coordination. The purpose of this chapter is to provide an understanding of the prior research in the fields of facial affect and gaze analysis, as well as presenting the rationales for the proposed studies.

2.1 Facial Affect Recognition

A common facial affect recognition system consists of three main components: face registration, feature descriptor and affect classification.

Face registration is to align the faces and minimize the non-expression variations. The face detector proposed by Viola et al. [109] is the most frequently used algorithm to locate the bounding box of a human face from an image. To further identify the facial landmarks, also called as fiducial points, active shape model (ASM) [22] and active appearance model (AAM) [21] are commonly used techniques. Some state-of-the-art techniques demonstrate promising performance of facial landmark tracking under diverse lighting conditions and significant facial deformations, notably the Constrained Local Model (CLM) [95] and the tracking with Supervised Descent Method [118].

Appearance and geometric features are two main categories of the facial feature representation. Holistic appearance features are extracted from the texture inside the face bounding box. Patch-based appearance features are obtained from the small patches around each facial landmark. The prevalently used descriptors include the Gabor Wavelet coefficients, Histogram of Oriented Gradient (HOG), Local Binary Patterns (LBP) and scale-invariant feature transformation (SIFT). While the appearance features are more sensitive to the subtle changes, geometric features require less computation, and they are more robust to the impact of shadow and intuitively interpretable. The common forms of geometric features include the coordinates of facial landmarks and the relative distances and angles between particular landmarks.

A variety of supervised machine learning algorithms have been applied in the research of facial affect recognition. McDuff et al. [77] compared the performance of generative and discriminative classifiers on assigning valence labels to facial action sequences. Littlewort et al. [65] evaluated AdaBoost [31] and Support Vector

Machines (SVMs) [12] on recognizing of basic emotions. They also used SVMs to recognize AUs and expressions of posed and spontaneous pain [66]. Hoque et al. [41] explored detections of frustration and delight by applying SVMs, Hidden Markov Models and Hidden-state Conditional Random Fields. El Kaliouby et al. [55] inferred the cognitive mental states using dynamic Bayesian networks (DBN). Li et al [62] applied DBN to estimate the intensity of AUs.

A comprehensive investigation of spontaneous facial expression recognition can be found in Zeng et al. [121], Sariyanidi et al. [97] and Corneanu et al. [23], and the recent challenges like AVEC [106] and EmotiW [25]. With few exceptions, most of the previous efforts are based on supervised learning, which requires intensive manual labeling of the facial data.

2.2 Reducing Human Effort of Data Annotation

To reduce the annotation effort, previous work has investigated various degrees of supervision for facial affect modeling.

Compared to the conventional supervised learning that requires a huge amount of frame-level annotations, some notable research work proposes to annotate the most informative frames only. Zhang et al. [122] used an interactive technique that initializes the affect labels with Bayesian networks and then used mutual information to select informative data for human correction. The goal is to label only the most optimal data. Zhu et al. [129] used dynamic cascades to identify frames that are proximal to the apex between onset and offset to increase the amount of training data for AUs detection. De la Torre et al. [60] labeled only the apex of the AUs and automatically predicts the corresponding onset and offset. However, the above approaches all require human expertise to locate and label some essential data, such as the apex frames, which is time-consuming and expensive, and probably infeasible for user-dependent modeling.

Apart from labeling the particular frames requiring expertise to identify, some studies propose annotating the average frame (i.e. the centroid) of a group of similar frames. Unsupervised learning uses clustering to identify similar facial expressions or facial gestures. De la Torre et al. [59] proposed a geometric-invariant clustering technique that segments a specific user's facial behavior into facial gestures. Zhou et al. [127] used Aligned Cluster Analysis to detect facial events from video across multiple individuals. Both approaches identify similar expressions across different

users. These methods can successfully discover similar facial events/gestures, but they do not aim to correlate the gesture with the affect, or address the differences in exhibited expression and felt affect across individuals. A good number of affect implications still needs to be confirmed by human experts or the self-report.

Weakly supervised learning based on coarse-grained segment-level annotation, rather than the fine-grained frame-level annotation, has been attracting recent research attention. Xu et al. [119] divided a facial sequence into 20 representative sub-motions based on optical flow, and applied "bag of motion words" to recognize basic emotions in the facial sequences. Their work targets sub-motions at the facial gesture level, which last around 100 frames (*i.e.* 4 seconds). However, manual annotation at such granularity in the client (target user) side is still infeasible. In real-use situations, a segment should be long enough to present a natural expression and short enough for a singular emotion, which can be approximately 1-2 minutes as suggested by the video lengths of the public facial datasets [101][46]. There are some promising research efforts that aim to learn from weakly labeled data using multiple-instance learning (MIL) techniques.

2.3 Learning Facial Affect from Bag Annotation

Multiple-instance learning refers to machine learning approaches in which a set, or "bag", of instances shares a common overall label, or a "bag annotation". In the context of facial affect learning from video, an instance is a frame from a video segment and a bag is the segment itself. Recent studies generally follow three main approaches when learning from the segment annotation. The first approach assigns all the instances in a bag with the bag label. Viola et al. [110] developed a boosting variant called MILBoost, which initializes all the instances (e.g. individual frames) with the label of the bag and applies boosting for further learning. Sikka et al. [99] extracted facial gestures from a video segment and employed MILBoost for pain recognition from facial expression. These methods assume that a large proportion of the instances in a bag coincide with the annotated label of the bag. However, this assumption may not hold in real-use facial affect recognition systems, as it is not uncommon to have expressions with different affect implications to manifest within a given segment of time in natural contexts.

Rather than use all the instances in a bag, the second approach adopts a subset of them as representation. For instance, Ashraf et al. [6] proposed to cluster the facial expressions in each segment and use the centroids to represent the segment for pain

detection. However, the same problem occurs: when mixed emotions are present, and with some emotions that are more momentary in nature (e.g. surprise), the affects exhibited by some centroids may not be consistent with the segment annotation.

The third approach devises a new feature space to characterize a bag. Chen et al. [15] determined bag similarity based on bag-to-instance distances. All instances are used to form the bag-level feature vector. However, this generates a high-dimensional space. Fu et al. [33] simplified the prototype vector by selecting only one instance per bag, which generates a vector with much lower dimension. Xiao et al. [117] explicitly measured the bag dissimilarity taking into consideration the instance similarity between the positive and negative bags. However, since the bag similarity is defined as the pairwise distance between the instances, the computation exponentially increases as the number of instances and bags. Cheplygina et al. [16] studied different forms of prototypes to measure the bag dissimilarity, including representations at instance-level and bag-level. They then proposed a balanced method using random subspace as the prototype. Despite their success, using selected instance(s) for bag description may not be suitable for facial affect recognition, notably for lengthy segments with a variety of instances.

Other efforts particularly work on facial affect recognition in the MIL paradigm. Ruiz et al. [92] proposed to identify multiple prototypes through an optimization mechanism, which jointly learns the prototypes and the parameters for the bag classifier. However, their study has not suggested the way to determine the number of prototypes. Additionally, since the prototypes and the classifier are jointly determined by the training set, this method may require a computationally expensive optimization for each model update with newly collected data. To reduce the computational cost, Huang et al. [47] encoded the segment characteristic by the probabilities of different emotional frames, but the identification of these emotional frames is still highly constrained by the proportion of positive instances in the training bags.

In line with the previous research that makes use of the potential indicative instance, we proposed an association-based multiple-instance learning (AMIL) [46] technique to ascertain the indication of facial gestures from their distributions across segments with different annotations. For the simplicity, we firstly explore the affect recognition depending on a voting mechanism of AMIL results. As a further study, we also cast the facial affect recognition problem into a classification problem to interpret the segment features. Compared with the previous methods, our approach can efficiently adopt the

indicativeness of important instances and describe the overall distribution of instances inside a segment in a simple but effective fashion.

2.4 Personalization for Affect Recognition

Much current research in affective computing focuses on model generalization for new users [54]. However, generic, or user-independent models have difficulty accommodating individual differences. Littlewort et al. [65] reported an accuracy drop from 95% to 60% when a model trained on one dataset is tested on another. Michel et al. [78] carried out similar experiments and the accuracy drops from 87.5% to 60.7%. Findings from the first facial expression recognition and analysis (FERA) challenge [107] also show that the user-dependent model generally outperforms the user-independent model.

There have been efforts in combining source and target data into the same model. Valstar et al. [107] showed that high performance could be achieved for emotion recognition when the prior training data for the target user is available. However, well-labeled user-specific data is expensive to obtain for real-use systems. There is also a data skew issue in the direct data aggregation. The contribution of the target data is likely to be overwhelmed by the much larger source data.

In spite of the previous success of automated facial affect studies in lab scenarios, recognition of spontaneous expression in natural contexts is still challenging [46]. An ideal solution to distinguish subtle expression differences under inter-personal variations is to personalize a user-specific model. Transfer learning has recently become popular for knowledge adaptation and addressing the target data scarcity issue in different problems [82], including document, sentiment, and image classification. For example, Dai et al. [24] extended Adaboost [31] for inductive transfer learning, which assumes some annotated target data is available. Other studies also investigated the transductive transfer learning, which assumes the target data is available but not labeled [13].

Despite the success of transfer learning, there has not been many efforts into applying it for facial affect recognition, nor has there been much attention on addressing interpersonal differences. Chu et al. [18] showed the effectiveness of the transductive transfer learning approach, which re-weighs the source training samples most relevant to the target user. However, instance re-weighting and model adaptation require a

computationally expensive optimization. Sangineto et al. [94] presented an alternative method to mitigate the computation of learning a personalized model, by using a regression function to identify the target model parameters based on the mapping from source subjects' data distribution into their classifier parameters. Zen et al. [120] further simplified the mapping by using support vectors for parameter transfer. Additionally, compared to the transfer learning from single combined source [24] (i.e. a combined set of data of all source subjects), learning from multiple single-sources [94][120] has a higher chance of identifying similar users, which should achieve a better transfer. However, the approaches that use the target data without considering the annotation may fail to achieve a correct adaptation. Chen et al. [13] compared transferring knowledge with and without using target data annotation. They demonstrated that the inductive transfer learning with target data annotation outperformed its counterpart.

Personalization of facial affect at the segment level can be even more challenging, due to the uncertain and subjective connection between the overall affect label and a video segment (usually thousands of frames), and the inadequacy of annotated target data (e.g. only dozens of annotations per subject). Therefore, transferring from a set of individual models [94][120][13] may not be a good choice in real-use situations. Since the amount of one subject's training data is usually insufficiently small, each of these weak individual classifiers may suffer from overfitting.

A large body of prior work has been done on the generic classifier [97]. Recent research suggests knowledge adaptation is effective to accommodate individual differences [94][18][120][13] and multiple-instance learning useful to reduce annotation effort [99][6][92][47].

This thesis proposes a framework to jointly solve these two issues. We use a variant of bootstrapping to transfer knowledge from groups of source subjects. While attempting to accentuate the weights of source data that is similar to the target user, we wish to maintain sufficient diverse data for each weak classifier. In contrast to learning from a set of individual classifiers, we coordinate a set of weak generic classifiers, each of which is learned from data of a subset of the source subjects.

2.5 Webcam-based Gaze Learning

Gaze estimation methods can be categorized into appearance-based and model-based. Model-based methods (e.g. [45][116][17][128]) build the mapping between the gaze point and eye geometric features such as location of pupil center and the contour of iris,

while appearance-based methods (e.g. [68][70][102][69][123]) map the image of eye regions to gaze point without explicitly extracting the eye features.

Unlike the eye tracking system based on stereo vision or infrared sensing, the single webcam gaze system needs to overcome a variety of variations, such as eye-camera distance, head pose, and glasses occlusion. The only solution is to learn from the training data. Appearance-based learning generally generates high dimension of feature vector and the resulting model can be sensitive to illumination changes. To overcome these, Zhang et al. [123] explored using sophisticated models and large dataset to learn the appearance-based user-independent gaze model. Their work shows learning from a large amount of data can effectively alleviate the impacts of pose and illumination. However, the user-dependent model still outperforms the user-independent model. This motivates our research on continuous and non-intrusive data collection for the target user.

A good number of studies investigate the robust techniques of eye geometry extraction to address the impact of the illumination and occlusion. Zhu and Yang [128] used the vector from inner eye corner to the fitted iris center as feature representation. Cheung et al. [17] applied the same eye vector with the head pose information to accommodate the pose variation. In a similar spirit of considering both eye and head pose, Valenti et al. [105] proposed a pose-retargeted gaze estimation method based on the eye center displacement with respect to the head pose change. Wood et al. [116] calculated the optical axis based on the fitted ellipse of the iris limbus and estimated gaze by trigonometric approximation. However, simple rule-based eye geometric features extraction in the wild can be unreliable. Huang et al. [45] suggest a data-driven approach to identifying a set of eye and iris landmarks. However, training a reliable non-rigid tracking model needs numerous landmark annotations. Wood et al. [115] leveraged the computer graphics techniques to synthesize various eye appearances, and showed promising landmark tracking result by the model learned on the automatically generated annotations. Although prior research shows success in robust eye features extraction, sufficient data acquisition for user-specific gaze learning is still challenging.

2.5.1 Generalizing from limited data

Making good use of limited data is key to improving performance while reducing the calibration effort. Williams et al. [113] developed a semi-supervised method to use unlabeled calibration data. Their method requires users to follow an animated spot on

the screen. Lu et al. [68] introduced decomposition scheme to correct the head pose biases for gaze estimation. Their calibration requires the user to rotate his/her head while fixating on each calibration point. To reduce the amount of calibration needed, Lu et al. [69] synthesized training samples for unseen head poses from multiple reference images where the user's head position changes while the eye rotation is held constant. Wood et al. [114] used the graphical rendering technique to generate numerous realistic eye images for gaze estimation.

Despite the success of the learning based on data synthesis, generalization for large head orientation and distance variation based on only modest reference images is still a challenging issue. Although these methods reduce the amount of total calibration data, one drawback is the requirement of a specialized and explicit calibration procedure. In addition, human error during calibration, such as eye blinks [70] or distracted saccades, may cause an unexpected performance drop. It is not difficult to see that a method that implicitly collects good training data can also be used in conjunction with the above approaches to further improve gaze modeling.

2.5.2 Implicit collection of incremental gaze data

Some approaches bypass the cumbersome and lengthy calibration phase by implicitly collecting data from daily computer usages. One popular solution uses a saliency model that assumes the user is more likely to look at the salient region of an image or video frame. Sugano et al. [102] applied the saliency map of video frames to estimate gaze based on images captured by a monocular camera. The problem with this approach is that the consistency between image saliency and real gaze location is often influenced by the attributes of visual stimuli, such as complexity and semantics. Apart from these uncertainties, the computation saliency models often do not match the actual human gaze movement [53]. Alnajar et al. [2] therefore made a different assumption that infers the calibration of a new user based on previously collected gaze data from a group of individuals. This method makes use of interpersonal similarity for visual attention. However, the correlation between visual attention and image conspicuity is also affected by differences between individuals.

There has been some work into adopting interaction information to facilitate gaze learning. Hornof et al. [42] suggest a strategy that looks for interactions with known fixation points for run-time recalibration of the eye tracking model. Zhang et al. [126] further identified probable fixation locations to account for instances that cannot be

clearly mapped to a known fixation point. Their work, however, relies on an infrared eye tracker to detect eye fixations, and knowledge of the visual context, including target locations and layout irregularity.

Sugano et al. [103] proposed an alternative model which collected mouse click points as ground truth data to incrementally update the gaze model. Jacob [48] used click data to correct gaze tracking results. Similarly, Fares et al. [28] proposed to use mouse-click data as dynamic local calibration data. These approaches all assume that the location of the click point is the location of the user's gaze. However, in unconstrained real-use situations, this assumption may not always hold.

2.5.3 Investigating the gaze-cursor correlation

Gaze-cursor consistency is a perennially popular topic of study, especially for web browsing behaviors. Chen et al. [14] suggest that there is a strong correlation between gaze and saccade-like mouse movement. Rodden et al. [89] reported strong alignments between gaze and cursor during active mouse usages, including using the cursor as a reading aid (in both horizontal and vertical directions) and to mark particular results. Guo et al. [37] proposed a set of mouse features to identify the moments with strong gaze and cursor alignment during browsing. They achieved an average accuracy of 77%, 3% higher than the baseline. Liebling et al. [64] showed that gaze and mouse coordination contain complex and nuanced characteristics in real-life scenarios. Huang et al. [44] found that there is a certain correlation between gaze and cursor, but with substantial variation whereby the distance between the eye gaze and the mouse click location is smallest one second before the click occurs for one-third of the subjects, in a "cursor lags behind gaze" phenomenon [43]. These findings suggest that the conventional hypothesis that "gaze is well approximated by cursor" may be naïve. It also shows that temporal alignments varied significantly across individuals, which argues for a personalized approach.

2.6 Mental Stress and the Gaze-Click Pattern

Stress level sensing is a far-reaching research topic, due to its profound connections with human well-being, including emotion, cognition, and health [104]. The conventional stress measuring techniques rely on physiological signals, such as galvanic skin response, muscle tension and heart rate variability [74]. However, that

these approaches require special equipment, together with their intrusiveness, discourage the wide use of physiology-based stress detection *in-situ*.

There has been work on non-intrusive stress detection based on multimodal signals. These methods usually use visual cues such as facial expression and body movement, or audio information such as vocal tone. However, these signals are sensitive to environmental noise and are highly person-dependent. An alternative approach uses information from daily input devices, *i.e.* keyboard and mouse. Pentel [84] studied the human confusion pattern from the mouse log data. Hernandez et al. [40] investigated the linkage between user stress and force in keypresses or mouse clicks. Their method achieves promising results and has the advantage of being non-intrusive, but it requires the use of special pressure-sensitive equipment. Sun et al. [104] advocate measuring stress by modeling the "stiffness" of mouse movement trajectory using a user-dependent approach. These studies show that long-term monitoring of stress can be universally accessible and unobtrusive.

Using eye gaze for the inference of human mental state also attracts the recent research attention, due to non-intrusiveness and informativeness of gaze patterns. Jaques et al. [49] investigated the engagement detection using gaze information. Gingerish and Conati [34] used eye gaze behaviors for task performance prediction. Andreu-Perez et al. [5] proposed a useful tool for the analysis of a variety of gaze behaviors.

When paired with the gaze, cursor patterns provide indicative cues. As mentioned in the gaze-cursor studies, there are conflicting views on the gaze and cursor alignment. Rodden et al. [89] and Guo et al. [37] claim gaze can be well approximated by the cursor. However, Huang et al. [43][44] and Liebling et al. [64] suggest that the gaze and cursor alignment can vary substantially in not only searching tasks but also other daily human-computer interactions. This controversy suggests that some important hidden factors may have been overlooked in the previous research on gaze-cursor pattern analysis. It is a normal phenomenon that participants in the lab experiments may suffer from different degrees of stress or uncomfortableness. This mental factor can be an overriding cause for the inconsistency of the previous research findings. In spite of the fact that both gaze and cursor information can be non-intrusively accessed in common computer setups, and the plethora of studies on this topic, no prior work has explored the impact of stress on gaze-click behavior.

2.7 Summary of the Related Work

Inspecting from the prior related research, we see gaps between the existing techniques and a practical model that can be deployed in real use. The constraining issues include the learning mechanism from the self-reported annotation and the utilization of generic knowledge for the adaptation to the target user.

The review of the gaze literature also suggests opportunities of modeling visual attention from the daily personal interaction data and detecting mental stress non-intrusively from the gaze-click patterns.

These studies together facilitate the interpretation of human expression and visual attention, and therefore, provide a practical channel to further understand the human mental states.

Chapter 3

PADMA – Personal Affect Detection with Minimal Annotation

Notations in this chapter

A	the set of affects, $\{a_1, \ldots, a_{\lvert V\rvert}\}$, $\lvert V\rvert$ is the type number of self-reported affects
a_i	the *i-th* affect
$\boldsymbol{c}_i$	the *i-th* facial centroid feature vector
d_{ij}	Euclidean distance between two cluster centroids $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$
$\boldsymbol{e}$	resulting expression label sequence after run-length encoding
e_i	the *i-th* expression label in the run-length encoded sequence
G_w	the set of gestures occurring in the windows that span over the *w-th* element in the run-length encoded sequence
G	the set of all facial gestures
g_i	the *i-th* facial gesture
$\boldsymbol{J}$	jitter frequency matrix, whose elements represent the count of transition
j_{pq}	transition count between centroid *p* and *q*
K	number of initial cluster for K-means
$\boldsymbol{l}$	the sequence of expression labels for a segment, generated by replacing the facial feature vectors with their closest centroid ID
l_p	expression label *p*, i.e. the *p-th* centroid ID
m	the number of facial gesture
n	the number of frame in a video segment
W	the elements in the run-length encoded sequence
T_t	duration threshold to determine temporal jittering
$\boldsymbol{t}$	resulting duration sequence after run-length encoding
t_i	the duration of the *i-th* expression in the run-length encoded sequence
V	the set of all response clip-sets
$\lvert V\rvert$	the number of different self-reported affects
v_i	the response clip-set for affect a_i, a group of video segment labeled with the same affect
$\boldsymbol{x}_i$	the *i-th* facial feature vector in a facial response segment
x_{ij}	the *j-th* facial feature in the *i-th* facial feature vector in a facial response segment
λ	jitter frequency threshold

Here are the graphical illustrations of the terms that we used in this chapter:

Each response clip-set for a particular affect contains multiple response video segment:

A response video segment:

The corresponding overall self-reported affect (segment-level annotation):

The cued-recall result (frame-level annotation, one reported affect every 4 seconds):

Example facial gestures (frequent subsequences in facial centroid ID sequences):

This chapter describes a user-specific approach to facial affect modeling. In order to reduce the annotation effort of building a user-specific facial affect model, we investigate the validity of MIL techniques for facial affect modeling from video segments. To provide the foundation for the segment-level affect recognition, this chapter analyzes the fine- and coarse-grained facial behaviors and human perceptions reflected on the two facial datasets: UNBC and MSARV.

MSARV is a spontaneous dataset that we collected for the purpose of this thesis. We collected both the segment-level during-experiment self-report data and the frame-level post-experiment retrospective annotations for MSARV. Likewise, we investigate the segment-level pain observation and the frame-level pain facial indicators on UNBC.

To explore the indicativeness of facial behaviors on these datasets, we introduce AMIL, an association-based multiple-instance learning technique, to quantify the relation between facial gestures and human perceptions. The findings of the facial gestures' consistency with the overall human perceptions and its association-based numeric representation of affect indicativeness provide the solid support for the model designs of the user-specific and user-adaptive facial affect learning.

Based on the findings from the facial behavior analysis, we further explore the use of AMIL in facial affect recognition system. To tackle visual noise, a novel adaptive clustering method is designed to fit the video data into AMIL. Evaluations on two datasets demonstrate the proposed method can recognize a diverse variety of basic emotions, interest, boredom, and pain in a simple and effective manner.

The flow of this chapter starts with the introduction of MSARV and UNBC datasets, based on which the relation analysis between the frame-level annotation and the segment-level annotation will be discussed, followed by the system overview of the proposed Personal Affect Detection with Minimal Annotation (PADMA) method, and the descriptions of video data clustering, encoding, and the AMIL algorithm. We then present the results of the corresponding evaluation of facial affect recognition.

3.1 Mobile Spontaneous Affect Response Video Dataset

There are a number of existing datasets from previous work. Chu et al. [18] and Valstar et al. [107] were tested on GEMEP-FERA [107], which consists of posed (simulated) expressions from 7 actors. Chu et al.'s work [18] was tested on Extended Cohn-Kanade

(CK+) [71] and RU-FACS [7]. Although these two datasets contain a good number of subjects, the data for any one subject is limited: around 100 frames for CK+ and 2.5 minutes for RU-FACS. DISFA [76] is annotated with AUs rather than facial affects, and the individual data is limited, around 4 minutes for each subject. Likewise, BP4D-spontanous [124] provides only short segments and limited individual data. MAHNOB-HCI [101] and DEAP [57] have sufficient individual data, but they do not provide frame-level facial affect annotation.

MSARV is a dataset that was constructed in-house at PolyU. The dataset consists of 11 Asian test subjects (5 female, aged 21-56, M= 32.4, and SD=11.8). Most are university students and staff. MSARV is available for research purpose (download from http://chilab.comp.polyu.edu.hk/?page_id=813).

The characteristic of MSARV is that it contains segments of spontaneous facial affects, captured on a mobile device. Elicitation videos are presented to the subject, and the front camera of the mobile device is used to capture the facial response. This produces a response video at 480x640 resolution and 30 frames per second. In total, the dataset contains 817,080 frames. The resulting head poses exhibited in the dataset, as estimated by the face tracker, are pitch: M=5.6^{o}, SD=6.3^{o}; yaw: M=-1.3^{o}, SD=2.5^{o}; roll: M=3.6^{o}, SD=3.5^{o}.

The video segments used for emotion elicitation include amusing scenes from the comedy “Gags”, talks about popular technology from “Engadget”, academic lectures on advanced topics, sad scenes from “Grey's Anatomy” and “Les Misérables”, eye and ear surgeries, trailers from horror and ghost movies, and video clips depicting abuse of pregnant woman, children and elderly people.

We assembled different segments into two elicitation videos, each approximately 40 minutes and containing 25 short segments. The content of each video is selected to elicit the following affects: happiness (1’30”x3), interest (1’8”x6), boredom (1’15”x5), sadness (2’19”x3), disgust (2’13”x2), fear (2’14”x3) and anger (1’32”x3). (The numbers in the brackets indicate the average length and the number of segments.) These are the same affects that are covered in most of the publicly-available datasets [107][71]. Some of the affects (e.g. happiness) are more easily aroused than others (e.g. sadness) [121], which accounts for the difference in the length and number of the elicitation videos. Segments are kept between 1-2 min to avoid habituation to the stimuli while being long enough to arouse an affect [35].

The experiment was performed in a private area (a research lab). Each subject was randomly assigned to watch one of the elicitation videos. Subjects were instructed to behave naturally and knew in advance that their expressions were being recorded. None of the subjects reported feeling inhibited with their emotions during the experiments.

Even though the elicitation videos were carefully chosen to elicit a particular affect, it does not mean that the viewers will necessarily feel that affect when viewing it. Therefore, after each video segment, subjects are asked to select an affect label (happiness, interest, boredom, sadness, disgust, fear, anger, or none/neutral) that best describes their *overall* feeling while watching the video segment. This is the limit of human annotation required in our approach. It takes only 1~2 seconds for each video segment and no particular expertise. They are used as segment-level ground truth to evaluate our weakly supervised learning approach.

3.2 Inspecting from Fine-Grained Facial Behavior and Overall Human Perception

To obtain the frame-level ground truth for MSARV, we followed a cued-recall procedure [91], which requires the subject to recall the felt affects from memory by the provision of visual information. This retrospective affect-judgment has been validated [91], proved to be consistent with external observations, and successfully used in previous work engagement detection.

The response video was synchronized with the corresponding elicitation video. The subject then watched the videos, together with two observers. Every 4 seconds, the subject and the observers were asked to label the response expression in the current frame with one of the affect labels in MSARV. If the subjects' self-evaluation is consistent with the observers' evaluation, the corresponding affect label is accepted as ground truth. In cases of disagreement, the subject, and the observers discussed until a mutual agreement was achieved. In addition to the 7 affects that we focus on in this work, facial expressions with no particular affect-related indication are marked as neutral. Table 3-1 summarizes the details. We show the length and number of the elicitation videos, the user response and the ground truth labeling, with respect to each affect. As indicated in the table, not all the videos successfully induced the expected affect, since the percentage of the self-reported affects and those of the elicitation video do not match perfectly. It turns out that interest and boredom have a much larger

Table 3-1. MSARV video details: elicitation videos, user responses, and manually annotated ground truth labels.

Affect	Length of elicitation video for each affect (sec)	Number of elicitation video segments for each affect	Number of response videos self-reported to be exhibiting each affect	Percentage of frames exhibiting each affect in the ground truth
N	0	0	1	3%
H	270	3	32	9%
I	408	6	87	27%
B	375	5	82	29%
S	417	3	13	6%
D	266	2	20	10%
F	402	3	16	7%
A	279	3	24	9%

N: neutral, H: happiness, I: interest, B: boredom, S: sadness, D: disgust, F: fear, A: anger

number than other affects. Surprisingly, neutral was rarely reported in our experiments. Post-experiment interviews suggest that this is because interest and boredom were available as options and users who did not feel that any of the basic emotions applied to them tended to choose one of those two affects instead.

Figure 3-1 presents the detailed relation between the segment- and frame- level annotations on MSARV. Each subfigure shows the frame-level annotation of different affects in segments with a particular affect annotation. The height of the bar shows the average percentage (or probability) of the frame annotation and the error bar denotes the range. For example, the probability of neutral frame in happiness segment is given by the average count of neutral frames in happiness segments normalized by the total frames in all happiness segments.

A closer scrutiny on the different frame percentages in a segment conduces to the understanding of the segment-level human perception and guides us to recognize the overall affect exhibiting in segments. As the data shows in Figure 3-1, the majority of the frame annotations are in accordance with the segment annotation. On average, near three-quarters (71.5%) of the frame annotations are in line with the segment annotations. However, the percentages also indicate that the occurrence of multiple affects in one segment is not uncommon. Seven out of 8 segment affects contain more than 3 types of frame affects, each of which has at least 10%. Some affects may even share a close frame proportion to the core affect. Specifically, there is only marginal differences between the average frame percentages of happiness (45.9%) and neutral (43.4%) in

the happiness segments. Similarly, the interest segments contain mainly the interest frame (51.8%) and also a good number of neutral frame (34.8%).

Although the overall frame- and segment- level human perception agree quite well, Figure 3-1 also indicates their occasional non-consistency. Observing the extreme cases from the error bars, we can see that frame-level recalled perception can be very different from the overall segment-level feeling. This is especially notable for the interest-related data. For example, there is a high percentage of "interest" frames in segments labeled

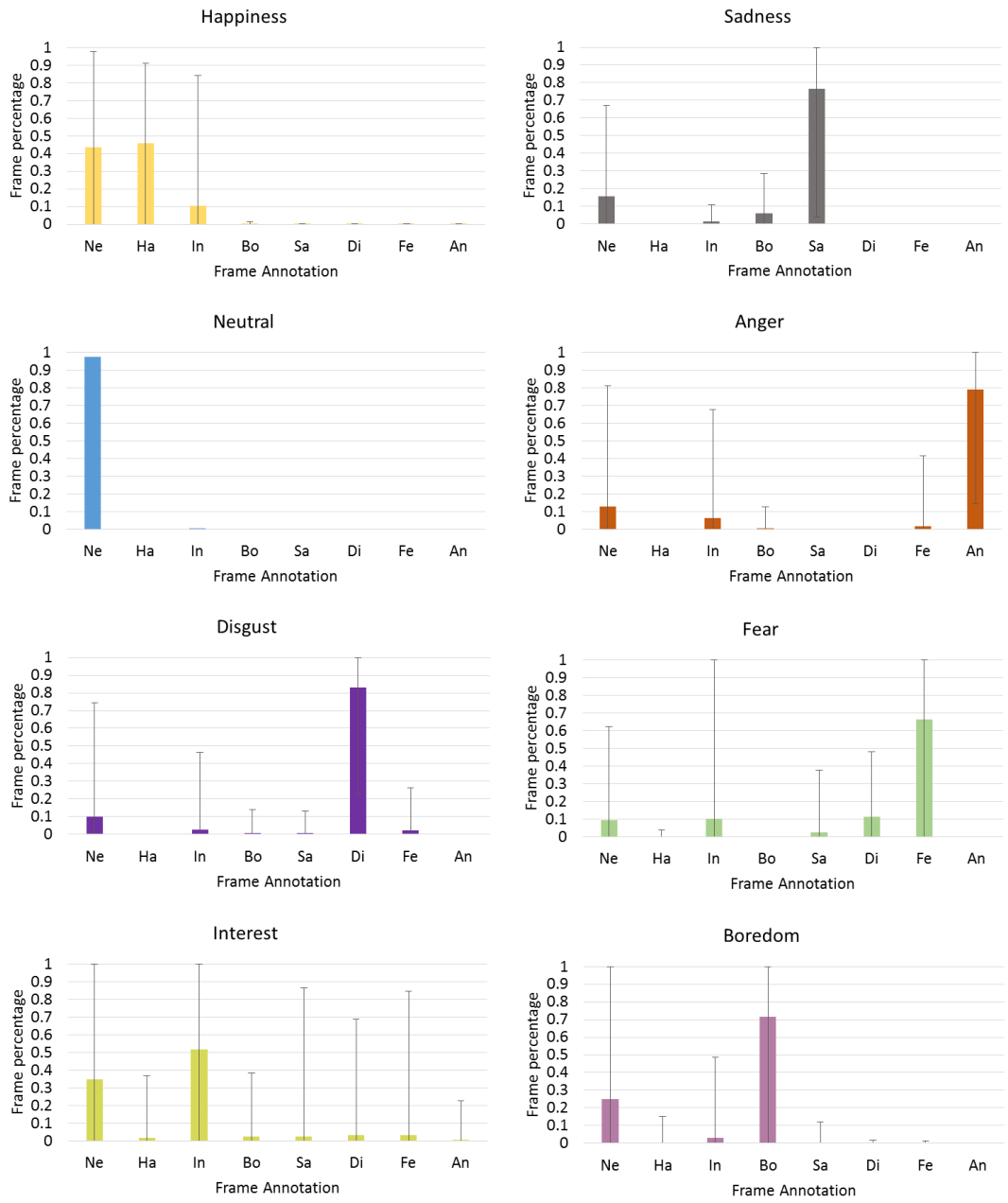


Figure 3-1. Percentage of frame-level affect in the segments with a particular segment-level affect on MSARV. The color bar shows the average percentage value and the error bar the range across segments.

as fear (100%), happiness (84.3%), anger (67.8%), disgust (46.2%) and even in a seemingly inverse affect, boredom (48.6%). It is also worth noting that the 100% frames being labeled as interest in a fear segment should be due to the human inconsistency between the during-experiment self-report and the post-experiment cued-recall procedure. It is highly likely that the subject felt a minor degree of fear after watching the elicitation video and reported so, but s/he thereafter failed to recall any obvious moment of fear from the facial expression during the cued-recall frame annotation. In addition, the frame-segment perception inconsistency in interest segment contains most diversity. There are some segments reported as fear in which more than 70% of the frames are labeled as neutral, sadness, disgust, and fear. This occasional, conflicting results between segment-level self-report and frame-level affect recall indicate that human self-perception of affect may vary across time and temporal granularity, particularly for the subtle spontaneous affects. This inconsistency makes the spontaneous facial affect recognition very challenging.

Neutral, as expected, appears in all kinds of segments. The average percentage of neutral frames across all non-neutral segments approach a quarter (21.6%), while in the special cases, neutral frames can be a majority, reaching 67% in sadness, 81.2% in anger, 74.6% in disgust, 62.4% in fear, and 100% in interest and boredom. It is worth noting that these neutral frames can, but not necessary, correspond to the expressionless facial expression. There are a variety of non-indicative, emotionless expressions with noticeable facial changes from expressionless expressions, such as the nonconscious facial movement caused by itchiness and irregular behaviors like swallowing saliva.

Under a close scrutiny, an interesting fact shows that some particular frame affects tend to occur in accordance with specific segment affect but not others. For example, sadness never appears in segments labeled anger or happiness; fear seldom shows up in happiness nor sadness; disgust happens only in segments of disgust, fear and interest segments. Such frame-level inclusiveness and exclusiveness can facilitate the segment facial affect inference from aggregating individual facial gestures (the fine-grained data).

It is also inspiring to observe that fear and interest segments contain frames with relatively diverse affect annotations. The diversity of interest is in line with the aforementioned post-experiment interview. Experiment participants preferred to choose either interest or boredom rather than neutral for the video stimuli. This may also be further explained by the point that interest is not a basic emotion. People can

become interested for dissimilar emotional reasons, resulting in different kinds of interest, such as happily interested, sadly interested and disgusted but interested. However, surprisingly, we perceive a good number of disgust frames in fear as well. This may due to the fact that the fear elicitation materials we used include a certain degree of disgusting factors, such as ghosts and zombies.

We compare facial behaviors on MSARV and another dataset, the UNBC pain-based dataset. We are interested in investigating the facial behavior on UNBC for two reasons. First, UNBC contains both segment and frame-level annotation from multiple subjects. This facilitates us to understand the facial behaviors in different granularity. Second, most prior MIL research [99][15][92] was evaluated on the UNBC dataset [72]. Evaluation on UNBC simplifies our comparison with the state-of-the-art performances.

The UNBC dataset contains 200 segments from 25 subjects with shoulder pain. Subjects performed active and passive arm movement with their affected and unaffected limbs. Expert coders gave Observer Pain Intensity (OPI) rating for each segment, ranging from 0 (no pain) to 5 (strong pain). We follow previous work [99][6][92] and define the segment-level label according to OPI, *i.e.* OPI≥3 is labeled as "pain" and OPI=0 as "non-pain", and intermediate intensities of 1 and 2 are omitted. Selecting subjects who have more than one video segment in the dataset gives us 147 segments and 23 subjects.

For the UNBC frame-level pain annotation, we follow previous work and determine the label according to the Parkachin and Solomon pain intensity (PSPI) [87]. PSPI measures the pain intensity from four AUs, i.e. the brow lowering (AU4), orbital tightening (AU6 and AU7), levator contraction (AU9 and AU10) and eye closure (AU43). And the corresponding frame-level AU annotation is given in UNBC.

Figure 3-2 presents the percentage of frames with different pain intensity in the segment-level pain and non-pain video clips. The x-axis shows the PSPI values, i.e. the pain intensity. The y-axis shows the frame percentage of a particular PSPI value in the pain (in orange) and non-pain segments (in blue). As expected, the majority of the frames in the non-pain segments have very low PSPI values (overall 97.5% of PSPI=0 and only 2.5% of PSPI=1). Although the frames in the pain segments have different degrees of PSPI, ranging from 0 to 12, a clear majority (76.7%) of those frames have a PSPI value of 0. This may seem surprising, but it makes sense that perception of pain, even for a short period, takes a dominant role in the entire segment.

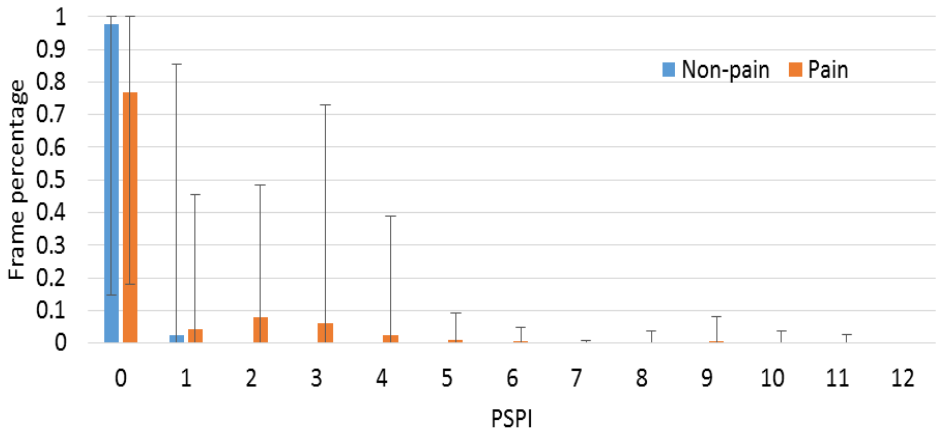


Figure 3-2. Frame percentage of the pain intensity in the segment-level pain or non-pain video clip. The color bar shows the value of the probability and the error bar the range across segments.

In summary, inspecting from the frame and segment annotations, it is common that a segment contains frames with different affect implications. The mixed affective states can occupy a significant percentage in a segment. Taking the happiness segments and the pain segments as examples, the annotated label of a substantial number of frames may not agree with the segment annotation. This supports our statement that directly assigning the segment annotation for every frame may not be proper. It is, however, encouraging to see that the clear majority of the frame annotations are still consistent with the segment annotation in most of the affects, indicating a simple majority voting may work to infer the segment-level affect from the fine-grained knowledge. However, the UNBC data presents an exception that under some contexts, an essential small proportion of indicative data may be determinative for an entire segment. To further understand the consistency of facial behavior with the coarse-grained annotation, we introduce PADMA and explore the relation between segment annotations and facial gestures in the following sections.

3.3 Designing the framework for User-Specific Facial Affect Modeling

Figure 3-3 illustrates the PADMA process on MSARV dataset. Similar to De la Torre et al. [59], we consider facial expression and affective state to correspond at the level of temporal facial gestures. That is, a change in the affective state results in a change in the user's facial expression, which is captured as a sequence of expression labels.

We start with an affect elicitation process to obtain samples of spontaneous facial affect from the user. Following Gross et al. [35], our affect elicitation uses video clips selected to arouse specific affects. The clips are chosen to arouse only one user affect at a time. To verify that the affects were elicited, users are asked to select an affect (including "Neutral" and "None of the Above") that best represents their overall feeling

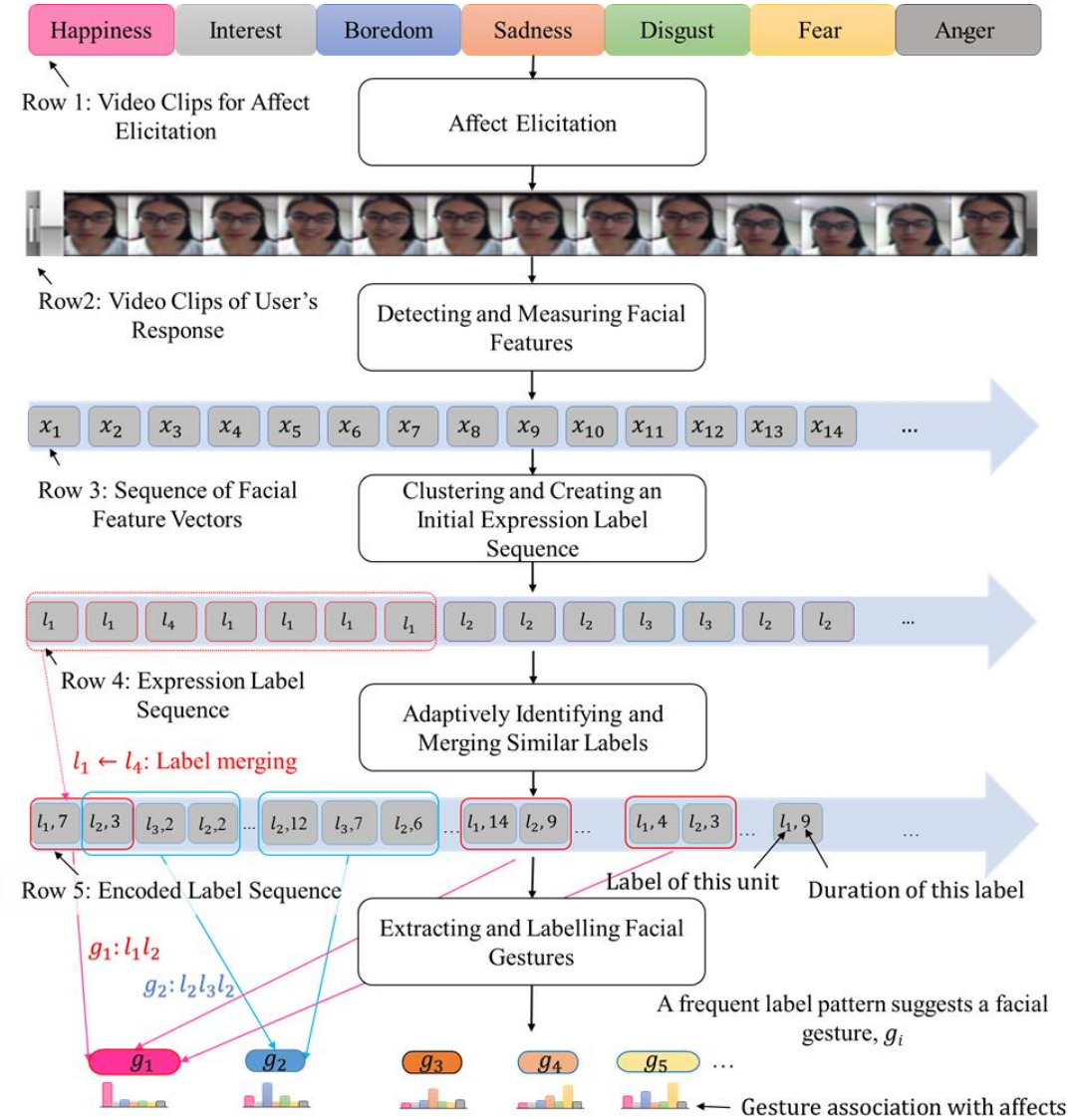


Figure 3-3. Personal Affect Detection with Minimal Annotation. Feature vectors (row 3) are extracted from the response video (row 2) to the stimulus (row 1). These vectors are clustered to create initial expression labels, which are then used to label the corresponding frames to create expression sequences (row 4). An adaptive merging process combines similar clusters (row 5) and facial gestures are extracted. Association-based multiple-instance learning is then used to determine the relation between facial gestures and affects depending on the occurrences of the gestures in segments and the corresponding self-reported affect labels.

after watching each video clip. In this work, we focus on 5 basic affective states and two higher-level mental states (Figure 3-3, Row 1).

We capture the user's facial response during affect elicitation (Figure 3-3, Row 2). Each frame in the user response video is processed to extract the facial features, which are then combined into a feature vector. This produces a sequence of facial feature vectors (Figure 3-3, Row 3). Clustering is then applied to group together feature vectors similar to each other. For each cluster, a distinct cluster label, or *expression label*, is introduced and used in place of the feature vectors to characterize each of the frames in this cluster (Figure 3-3, Row 4). This frame-labeling procedure, similar to vector quantization, encodes the large number of possible feature vectors as a relatively small set of labels to simplify and facilitate facial expression analysis in the subsequent steps.

A facial expression, with its onset, apex and offset, can therefore be represented as a sequence of expression labels. Run-length encoding and cluster merging are then applied to identify frequently-occurring expression label sequences, or *facial gestures*, from the response sequence (Figure 3-3, Row 5). Given the user's self-reported affects from the elicitation process, we infer the affective state that is expressed by each facial gesture by analyzing the distribution of facial gestures across the entire sequence of the response video and the correlation between the facial gestures and the affects (Figure 3-3, Row 6). Once the correlation is identified, the affective state of a user at any given point in time can be identified by looking for facial gestures that occur around that time period.

3.4 Modeling the User-Specific Facial Model

3.4.1 Detecting and measuring facial gestures

The affect elicitation process constructs a *response video* that contains the user's facial expressions for a given set of affects, such that these expressions may be detected and measured automatically.

Previous work in psychology and computer vision has proven the value of using AUs-based analysis for interpreting and analyzing facial expressions [65][66][77][55]. Facial AUs are descriptors of facial movements, which constitute the essential representation of a facial expression. Indeed, it is possible to describe all facial expressions as combinations of different AUs.

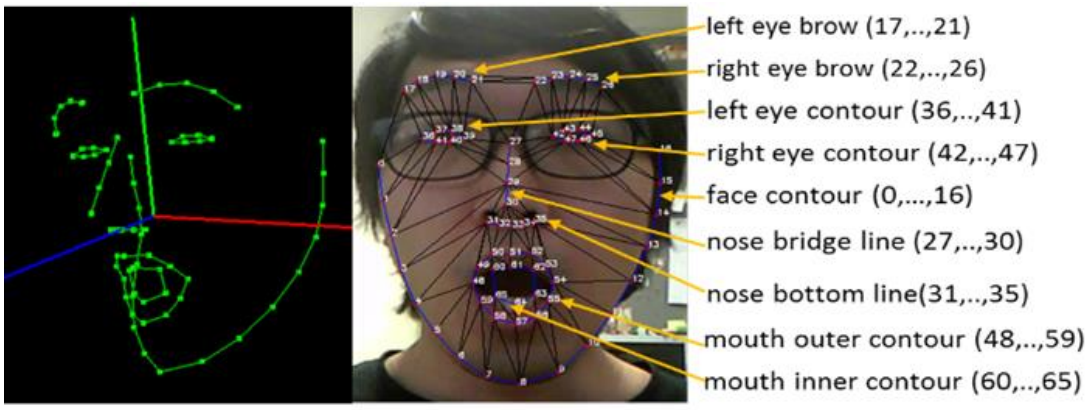


Figure 3-4. Facial landmarks tracked by CLM. The wired face (left) presents the tracked 3D facial landmarks. The facial image (right) shows the locations and indices (i.e. numbers in the bracket) of the corresponding 2D facial landmarks.

We follow an approach from previous work [59][127] to extract facial features referring to AUs. We apply Constrained Local Models (CLM) [95] to track 66 facial landmarks from the response video. This model is trained on the CMU Multi-PIE Face database [36], which contains over 750,000 images from 337 people. However, due to the nature of the training data, this model fails to track some of the mouth movements, such as mouth corner depression. To improve the tracking accuracy, we also leverage the Supervised Descent Method [118] to validate and optimize the 2D landmark locations. During the CLM optimization procedure, the 2D and 3D landmarks and other global and local parameters are adjusted iteratively until the face fitting regression model converges. Removing the rigid transformation from the acquired 3D shape compensates for the influence of out-of-plane rotation and produces the aligned 3D landmarks.

The direction and intensity of the facial movements can be calculated from the normalized distances and angles between the corresponding facial landmarks. This generates facial features that are similar to Motion-Units [19], which describe facial movement like Ekman's AUs; but are numeric and directional in nature, unlike AUs which are classified into discrete intensity levels.

The facial landmarks used in our work are shown in Figure 3-4. The wired face shows the 3D facial landmarks from the tracking result, and the facial image shows the locations and indices (i.e. numbers in the bracket) of the corresponding 2D facial landmarks. Table 3-2 presents the descriptions and measurements of the 20 facial features calculated from the aligned 3D landmarks.

Table 3-2. Geometric facial features used in our method.

Geometric features index	Implication	Measurement
1-4	Inner and outer brow movement	Distance between eyebrow corner and corresponding eye corners (left & right)
5-6	Eyebrow movement	Distance between the eye center and the corresponding brow center
7-8	Eyelid movement	Summed distance between corresponding landmarks on the upper and lower lid
9	Upper lip movement	Distance between the nose tip landmark and upper lip center landmark
10-11	Lip corner puller	Distance between the mouth corner and the corresponding eye outer center landmark
12	Eyebrow gatherer	Distance between inner eyebrow corners
13	Lower lip depressor	Distance between the chin bottom landmark and lower lip center landmark
14	Lip pucker	Perimeter of the mouth outer contour
15	Lip stretcher	Distance between the mouth corners
16	Lip thickness variation	Summed distance between corresponding points on the outer and inner contours
17	Lip tightener	Summed distance of corresponding points on the upper and lower mouth outer contour
18	Lip parted	Summed distance of corresponding points on the upper and lower mouth inner contour
19	Lip depressor	Angle between mouth corners and lip upper center
20	Cheek raiser	Angle between nose wing and nose center

Figure 3-5 shows sample frames from our experiment data. Both head movement and lighting condition (e.g. dissimilar illumination and camera exposure, etc.) pose significant challenges for the appearance-based features, especially with elderly people with natural wrinkles. Hence, to ensure robustness in real-use situations with various

environmental variations, we focus on geometric facial features, which avoids the noise from the textural/appearance channel.

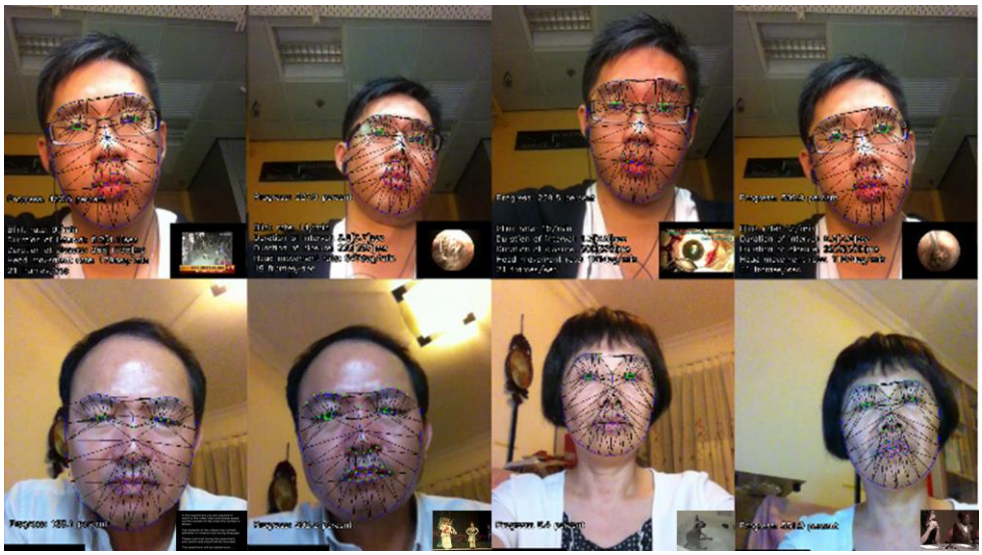


Figure 3-5. Example frames from our experiment data. The face tracking model is able to correctly locate the landmarks and gives precise geometric features, regardless of the lighting and facial appearance conditions.

3.4.2 Clustering and creating an initial expression label sequence

After the facial features are detected and measured, a user's facial response can be represented as a sequence of facial feature vectors, $\boldsymbol{x}_i = \langle x_{i,1}, \ldots, x_{i,20} \rangle \in \boldsymbol{X}$, where each $x_{i,j}$ is the measurement of feature j for a given frame i. This gives a quantified representation of the user's facial expressions in the response video, which is highly dimensional and difficult to manage. Dimensionality reduction is therefore used to render the changing user facial expressions more manageable.

We normalize each facial feature measurement of a user to a range between 0 and 1, then apply K-means clustering [75] to cluster together similar facial feature vectors. This allows us to identify *expression labels* for distinct facial expressions, which will essentially function as a low dimensional representation of the facial expression in the user's response.

Since the purpose of the expression labels is to represent different expressions, we need to avoid clustering together markedly different expressions. The cluster number

K is therefore chosen to be large (500 in our experiments). We perform a preliminary clustering on a random 10% subset of data to seed the initial locations of the cluster centroids. To compensate for the randomness in the clustering process, we repeat the entire clustering process 10 times and choose the result that gives us the most compact clusters, or the lowest intra-cluster distance, averaged over all clusters.

The centroids of the resulting clusters are then assigned unique IDs, which function as labels for the feature vectors. Each feature vector, $\boldsymbol{x}_i, i \in [1, n]$, is then replaced with the label for its corresponding cluster, *n* is the number of frames in a user's response video. This gives us a sequence of *expression labels* $\boldsymbol{l}$.

3.4.3 Adaptively identifying and merging similar labels

Theoretically, given the sequence of expression labels $\boldsymbol{l}$, identifying facial gestures should simply be a matter of looking for frequently occurring subsequences in $\boldsymbol{l}$. In practice, however, it is a challenge to decide on the number of clusters K used in the clustering process. A large K leads to redundant expression labels, where similar facial expressions are assigned to different clusters. This manifests as temporal jittering in $\boldsymbol{l}$, when the facial gesture sequence "bounces" back and forth between two labels over a short duration (Figure 3-6). A small K, on the other hand, may assign markedly distinct expressions to the same cluster, which may result in inadequate expression labels and the loss of potential indicative expressions.

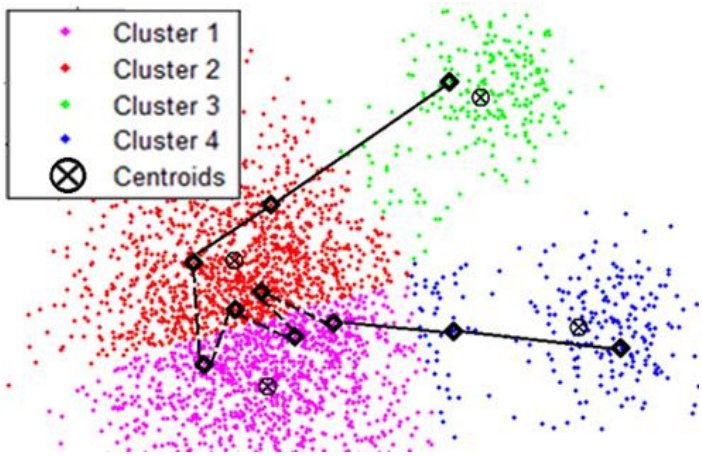


Figure 3-6. Temporal jittering caused by an over-large K value. Different colors indicate dissimilar clusters. Clusters 1 (purple) and 2 (red) are similar clusters that should be merged. The black lines denote an expression sequence. The dotted lines indicate jittering between cluster 1 and 2.

Table 3-3. Identifying jitters and merging expression labels

Input: expression label sequence of user's response $\boldsymbol{l}$
Output: expression label sequence with merged labels $\boldsymbol{e}$
a $(\boldsymbol{e}, \boldsymbol{t}) = \text{runLengthEncoding}(\boldsymbol{l})$
do
b $\boldsymbol{J} = \text{countJitter}(\boldsymbol{e}, \boldsymbol{t})$ via Equation (5.1)
c $\mu_i, \sigma_i = \text{calculateInterClusterDistance}(\boldsymbol{c}_i, \boldsymbol{c}_{j \neq i})$
d $\lambda = \mu_J + \xi \cdot \sigma_J$
foreach *pair* (l_p, l_q) **do**
e **If** $j_{p,q} > \lambda$ **then**
f $\tau_{p,q} = \min(\tau_p, \tau_q)$
If $d_{pq} < \tau_{p,q}$ **then**
g $(\boldsymbol{e}, \boldsymbol{t}) = \text{updateSequences}(\boldsymbol{e}, \boldsymbol{t}, l_p \leftarrow l_q)$
end
end
end
While *number of clusters being successfully merged > 0*

PADMA adaptively learns the proper number of clusters in a manner similar to G-means [38]. The underlying assumption is that changes in human facial expressions are usually continuous and progressive, and do not exhibit back-and-forth changes as would be suggested by temporal jittering. Since the jitter is caused when similar facial expressions are split into different clusters as a result of an over-large K, we merge similar clusters by minimizing temporal jittering in $\boldsymbol{l}$, subject to their distribution in the response video.

Table 3-3 illustrates the process for identifying jitters and merging clusters from the sequences of expression labels $\boldsymbol{l}$. Run-length encoding is used to decompose $\boldsymbol{l}$ into $\boldsymbol{e}$ and $\boldsymbol{d}$, where $\boldsymbol{e}$ is the encoded sequence of expression labels and $\boldsymbol{d}$ the frame duration of the labels. For example, $\boldsymbol{l} = \{l_a, l_a, l_a, l_a, l_b, l_b, l_c, l_c, l_c, l_c, l_c\}$ will be decomposed to $\boldsymbol{e} = \{l_a, l_b, l_c\}$ and $\boldsymbol{t} = \{4,2,5\}$ (Table 3-3, Step a).

We assume that a facial expression normally lasts for at least T_t frames, which we define as the duration threshold for expression label transition. We then identify a jitter by looking for all instances of u where the following conditions are fulfilled:

$$e_{u-1} = e_{u+1};\ e_u \neq e_{u+1};$$

$$t_{u-1} > T_t;\ t_u < T_t;\ t_{u+1} > T_t \quad (3.1)$$

For each jitter between two expression labels, l_p and l_q, we increment the corresponding entry j_{pq} in the jitter frequency matrix $\boldsymbol{J}$ (Table 3-3, Step b).

Simultaneously, we calculate the cluster distance between each pair of cluster:

$$d_{ij} = \| \boldsymbol{c}_i - \boldsymbol{c}_j \| \quad (3.2)$$

is defined as the Euclidean distance between the centroids of clusters $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$. μ_i and σ_i are then the mean and standard deviation of the distances between $\boldsymbol{c}_i$ and all other clusters (Table 3-3, Step c).

We define the jitter frequency threshold $\lambda = \mu_J + \xi \cdot \sigma_J$, where μ_J is the mean and $\sigma_J$is the standard deviation of all the nonzero data in $\boldsymbol{J}$, and ξ is a parameter that models the probability of jitter between two expression labels (Table 3-3, Step d).

If j_{pq} is larger than λ, then the clusters corresponding to l_p and $l_q$are potential candidates for merging. For each pair of such clusters, we calculate $\tau_{p,q} = \min(\tau_p, \tau_q)$, $\tau_i = \mu_i - \sigma_i/2$. (Table 3-3, Step f). If $d_{pq} < \tau_{pq}$, $\boldsymbol{c}_p$ and $\boldsymbol{c}_q$ will be merged (Table 3-3, Step g). The algorithm iterates until no more labels are merged.

After similar labels in the expression sequence have been merged, facial gestures are then identified by frequent subsequence mining in the multi-scale temporal moving windows across each segment based on the refined sequences of centroid IDs as in [46][110][99].

3.4.4 Association-based Multiple-Instance Learning

This section describes the AMIL technique used to explore the fine-grained data, i.e. the facial gestures, given the segment label. AMIL is an association-based method that is able to quantify the affect indicativeness of facial gestures, based on their distribution across segments with distinct affect implications. It is based on the process of data of each individual. Since AMIL learns from the sequences of expressions, it can be used in conjunction with different facial descriptors and clustering techniques.

Our measure is inspired by the tf-idf [52] measure used in information retrieval. We recast the problem of affect recognition as that of retrieving the most appropriate user affect, given a "query" of a facial gesture $g_i, i \in [1, m]$, where m is the number of facial gestures. We define v_i, the *response clip-set* for affect a_i, as the set of response video segments that were reported by the user as exhibiting affect a_i – that is, a_i is user-

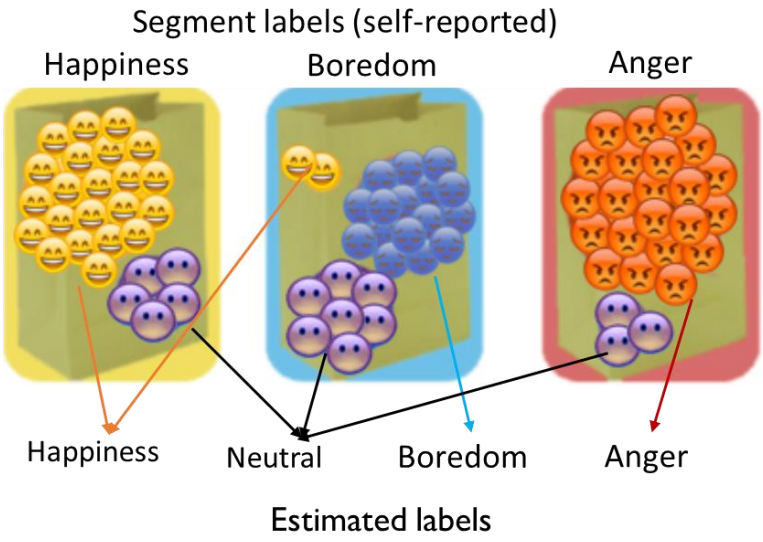


Figure 3-7. Analyzing gesture distribution. Different bag colors represent different self-reported labels for three response clip-sets. Emoticons represent facial gesture sequences. The size of the emoticon is proportional to the frequency of occurrence of the gesture. A gesture (e.g. purple) that commonly occurs across different bags is identified as neutral, while gestures that occur primarily in a particular clip-set are considered indicative of the affect associated with the bag.

reported to be the main affect experienced when viewing the corresponding elicitation clip. Therefore, we expect the facial gestures in v_i to exhibit mainly affect a_i and neutral, with a few other affects also included.

Figure 3-7 illustrates with an example. Three response clip-sets are shown, corresponding to the affects happiness, boredom, and anger, respectively. Facial gestures that occur almost exclusively in one response clip-set are identified as exhibiting that particular affect. On the other hand, facial gestures that occur regularly across multiple response clip-sets most likely are not representative of any particular affect, hence labeled as neutral. Therefore, our goal is to identify facial gestures that commonly occur in v_i, but not in response clip-sets for other affects.

We define $f(g_j, v_i)$ as the frequency of occurrence of the facial gesture g_j in response clip v_i. The *inverse affect frequency* (IAF) of a gesture quantifies the indicative value of a gesture by measuring its "rarity", on the basis that very common gestures have little indicative value:

$$\mathrm{IAF}(g_j, V) = \log \frac{1 + |V|}{\left|\{v \in V : f(g_j, v) > 0\}\right|} \tag{3.3}$$

V is the set of all response clip-sets; $|V|$ denotes the number of different self-reported affects, and $\left|\{v \in V : f(g_j, v) > 0\}\right|$ represents the number of response clips that

contain g_j. Since g_j represents an existing facial gesture in the response clips, the denominator is always nonzero. We set the numerator to be $1 + |V|$ to ensure the resulting IAF is larger than zero, so that it will not eliminate the contribution of other factors after the multiply operation.

The *response frequency* (RF), on the other hand, measures the prevalence of a facial gesture over the duration of v_i. Given the set of all facial gestures G:

$$\mathrm{RF}(g_j, v_i) = \frac{f(g_j, v_i)}{max\{f(g, v_i): g \in G\}} \tag{3.4}$$

$max\{f(g, v_i): g \in G\}$ denotes the maximum frequency of any gesture occurring in v_i, and normalizes bias towards longer response clips. The RFIAF value presenting the association between a gesture g_j and an affect a_i is calculated as:

$$\mathrm{RFIAF}(g_j, v_i, V) = \mathrm{RF}(g_j, v_i) * \mathrm{IAF}(g_j, V) \tag{3.5}$$

3.4.5 Labeling facial gestures and calculating facial affect

Denoting G_w as the set of gestures occurring in the windows that span over the *w-th* element in the run-length encoded sequence, we define the association between affect a_i and the gestures in G_w as follows:

$$R(\mathrm{G}_w, a_i) = \sum_{g_j \in \mathrm{G}_w} \mathrm{RFIAF}(g_j, v_i, V) \tag{3.6}$$

Identifying the facial gestures and associating them with the corresponding affect gives us a *description* of how a person expresses a particular affect. Given this information, identifying the affect is then a matter of looking for facial gestures.

Using the same multi-scale moving windows over each segment, we calculate the segment-level affect label a according to the $R(G_w, a_i)$ values across all windows:

$$a = \mathrm{argmax}_{a_i \in A} \sum_{w \in W} R(G_w, a_i) \tag{3.7}$$

where $A = \{a_1, \dots, a_{|V|}\}$ is the set of affects and W denotes the elements in the run-length encoded sequence.

Similarly, we can also estimate the frame-level label from the gesture-level estimation. For the *k-th* frame, the affect label is estimated by:

$$a^{(k)} = \mathrm{argmax}_{a_i \in A} R(\mathrm{G}_{\Phi(k)}, a_i) \tag{3.8}$$

Φ denotes the frame mapping from the original video sequence to the run-length encoded sequence.

We shall use the RFIAF values of the gestures occurring in a segment to present a clearer image between the facial gestures and the segment affect.

3.5 Understanding Facial Gestures Indicativeness by AMIL

On the basis of the affect indicativeness acquired by AMIL, this section presents the direct association results between the fine-grained facial behaviors and the coarse-grained human self-perception on MSARV and UNBC, which is measured by the RFIAF value. Since RFIAF represents simultaneously the prevalence of a gesture and its rarity across the affect clip-sets, a high RFIAF value therefore indicates a close association between the facial gesture and the corresponding affect.

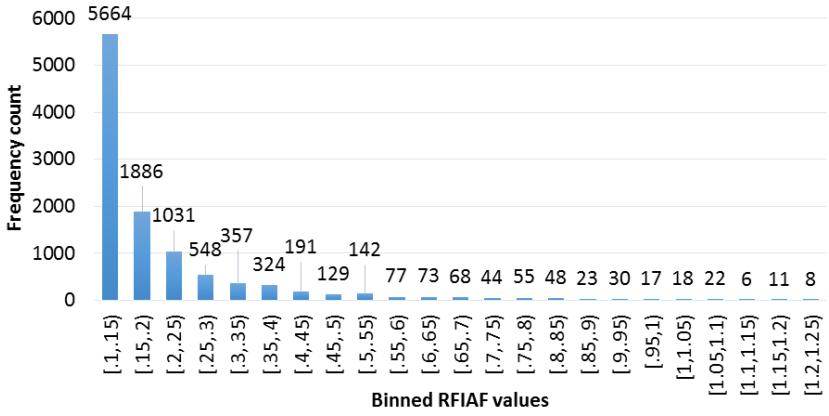


Figure 3-8. Histogram of the RFIAF values between facial gestures and all kinds of affects on MSARV. The size of the gesture pool with RFIAF value less than 0.2 is relatively large.

Figure 3-9 to Figure 3-16 show the histogram of the RFIAF of the facial gestures from MSARV. The x-axis denotes the binned RFIAF values and the y-axis the frequency count of the gestures in each bin. In these histograms, we remove the data with RFIAF value less than 0.2 for a clear presentation purpose, since these bins contain gestures that are basically non-indicative of any affect, and the size of the gesture pool in these bins is relatively large, as shown in Figure 3-8. We also truncate the bars with more than 100 counts to make the figure scale readable. Some example expressions across subjects in different bins are shown in the figures.

It is not difficult to see that the identified expressions indicate a correct implication of the corresponding affect. For example, the extracted facial expressions in the happiness segments reflect different kinds of smiles. Furthermore, it is encouraging that the expressions in bins with high RFIAF values tend to express obvious affects, and

those with low RFIAF values seem to convey subtle emotions. It can be seen that smiles with high RFIAF value show a strong sense of joy or amusement (e.g. (c) and (d) in Figure 3-9) and those with small values present are subtle ((a) and (b) in Figure 3-9).

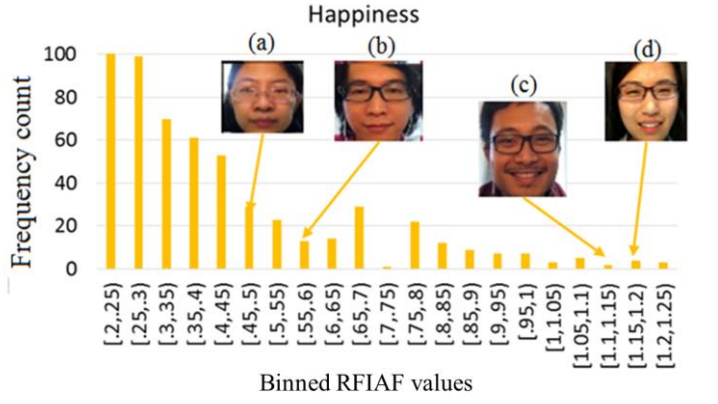


Figure 3-9. Histogram of the RFIAF values between facial gestures and clip-set labeled as happiness on MSARV. Some example expressions in particular bins have been given. Limited by space, only the key expression of a facial gesture is presented in the figure.

Expressions with high RFIAF in disgust present a clearer and stronger level of feeling disgusting, with more severe eye squeeze, frown and lip depressor (see Figure 3-10).

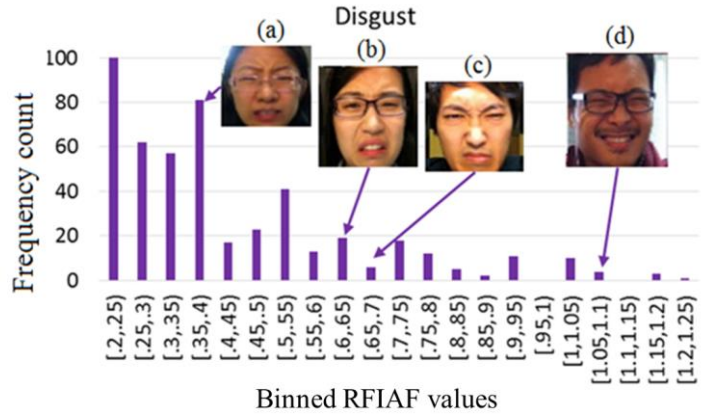


Figure 3-10. Histogram of the RFIAF values between facial gestures and clip-set labeled as disgust on MSARV.

A similar relation between RFIAF and the facial affect intensity changes also occurs in other basic emotions, including sadness, anger, and fear, though in a more subtle fashion, see Figure 3-11, Figure 3-12 and Figure 3-13.

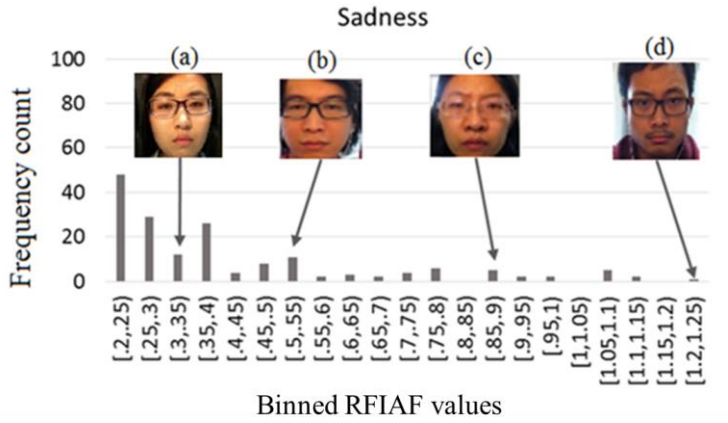


Figure 3-11. Histogram of the RFIAF values between facial gestures and clip-set labeled as sadness on MSARV.

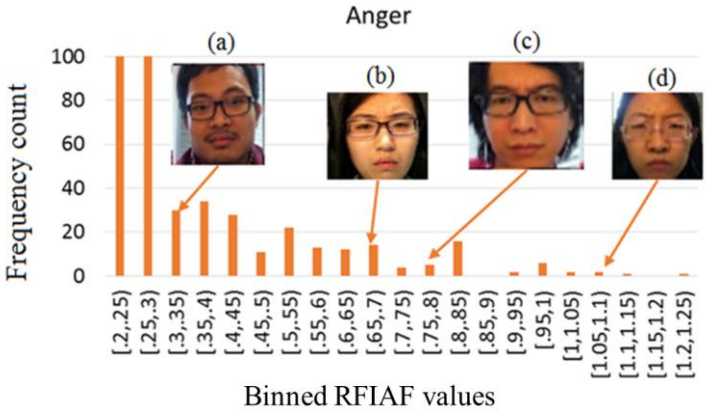


Figure 3-12. Histogram of the RFIAF values between facial gestures and clip-set labeled as anger on MSARV.

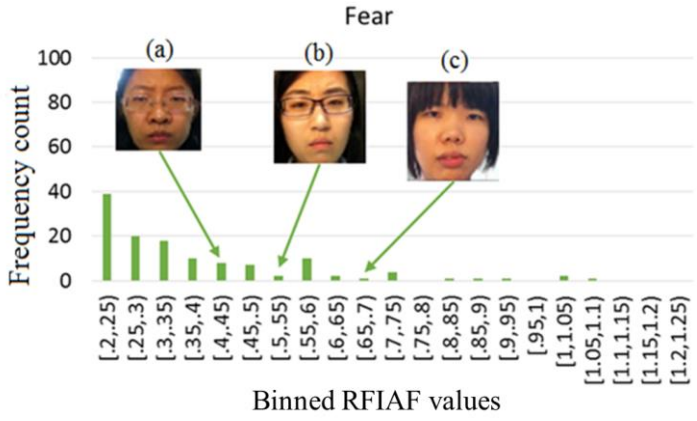


Figure 3-13. Histogram of the RFIAF values between facial gestures and clip-set labeled as fear on MSARV.

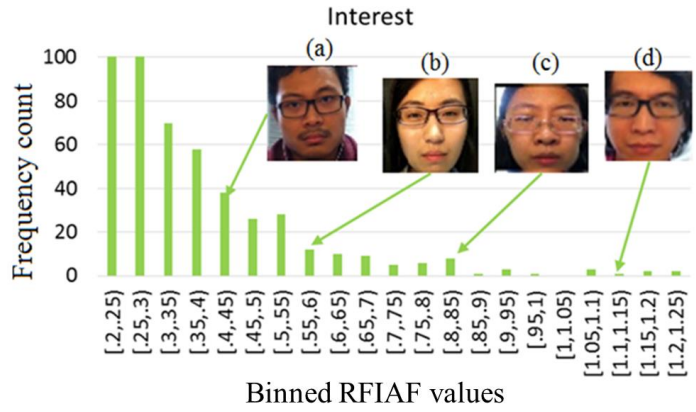


Figure 3-14. Histogram of the RFIAF values between facial gestures and clip-set labeled as interest on MSARV.

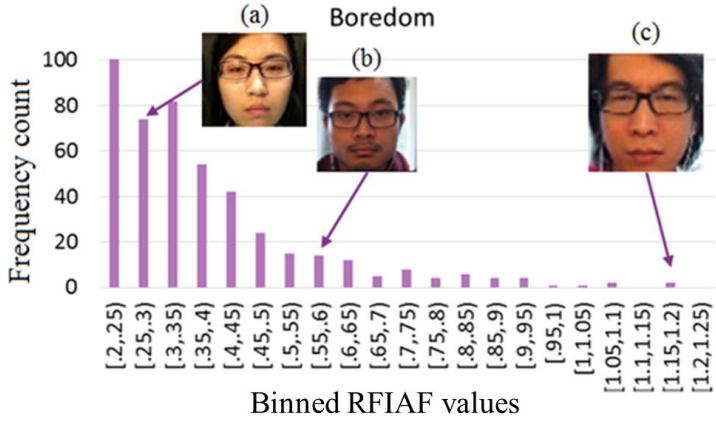


Figure 3-15. Histogram of the RFIAF values between facial gestures and clip-set labeled as boredom on MSARV.

Interest and boredom contain relatively high diversity, compared to the basic emotions. Some people present positive expressions when interested (see Figure 3-14a

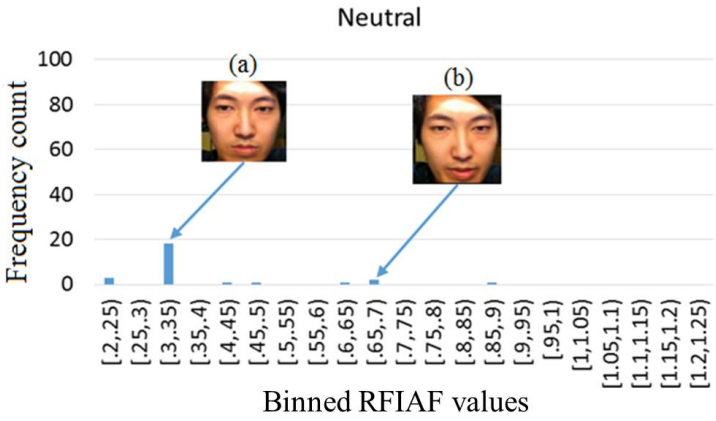


Figure 3-16. Histogram of the RFIAF values between facial gestures and clip-set labeled as neutral on MSARV.

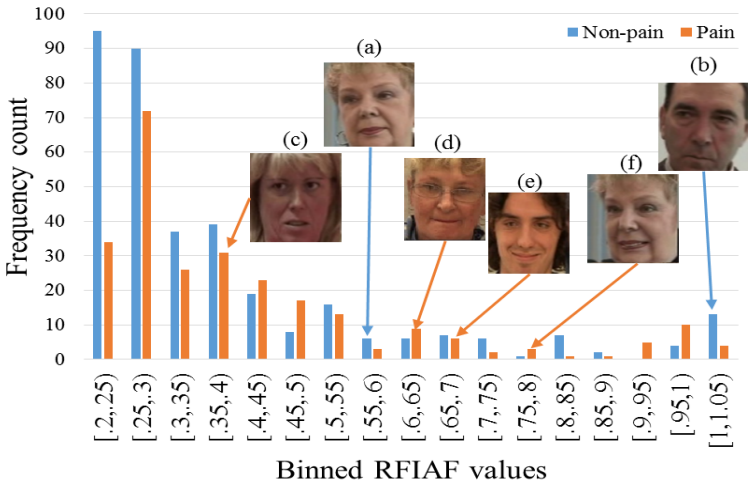


Figure 3-17. Histogram of the RFIAF values of facial gestures on UNBC. Blue bars indicate the gestures' RFIAF value with non-pain segments and the orange with the pain segments. Example expressions in particular bins have been shown.

(b)), conversely, some look more serious (see Figure 3-14 (d)). Similarly, some people express boredom by appearing neutral or sleepy (see (a) and (b) in Figure 3-15), while others look impatient (see Figure 3-15 (c)). Even given this diversity, the trend that higher RFIAF values correspond with more explicit affect implication suggests the effectiveness of using RFIAF to measure the association between facial gestures and affects.

In general, the number of indicative facial gestures decreases as the RFIAF value increases. However, differences still exist among affects. Some affects have less indicative gestures, such as neutral (see Figure 3-16), sadness (see Figure 3-11) and fear (see Figure 3-13). This may be explained by the elicitation effect during the experiment. Fear and sadness are more difficult to induce; and feeling neutral to the entire elicitation video is also uncommon. This elicitation difference is in good agreement with the data shown in Table 3-1, which shows that MSARV contains fewer such instances.

Apart from the differences in the number of indicative gestures, the dissimilarities among gestures in different bins are not linear across affects. For example, the identified gestures in happiness and disgust segments show considerable variation, while the facial gestures in the fear, sadness and interest segments, even with marked RFIAF differences, present only perceivably subtle changes.

Figure 3-17 shows the RFIAF histogram of the facial gestures and pain / non-pain on UNBC. The blue bars indicate the facial gestures' RFIAF value with non-pain and the orange with the pain.

It is not surprising that the extracted gestures with high association degree with the non-pain segments appear to be neutral expressions (see Figure 3-17 (a) and (b)). It is, however, interesting that the indicative gestures of pain are quite diverse for different individuals. For example, while suffering from pain, some people may show lip part (Figure 3-17 (c)), tightener (Figure 3-17 (d)), or even smile (Figure 3-17 (e) and (f)). Since AMIL extracts the indicative expressions based on the individual data, the impact of personal bias can be fully accommodated.

More detailed evaluation of applying AMIL to facial affect recognition will be given in the following section of evaluation.

3.6 Experimental Validation of PADMA

The contribution of our approach is a novel, weakly supervised method that uses AMIL to identify human affects from video data in real-use scenarios for user-specific affect modeling. It does not require expert annotation, nor does it require much human work for labeling. We shall validate its correctness and effectiveness in two aspects:

Contribution of our novel AMIL approach. AMIL differs from other MIL approaches by using an information retrieval-inspired approach that uses the distribution of a pattern across *all* bags for labeling. We evaluate the impact of this assumption against that of other MIL models, both at the segment (bag) level and at the frame level. For this purpose, we will reconstruct two high-performing MIL methods as representatives of current state-of-the-art [117], and compare the performance of our approach with theirs on a publicly-available dataset as well as our own dataset.

Overall performance of the PADMA method. We argue that a weakly-supervised user-specific model would be more appropriate in real-use contexts with spontaneous expressions. We will therefore evaluate PADMA against user-independent approaches. For a better understanding of the role of training data and user effort, as well as the advantages and disadvantages of each approach, we will also explore issues such as learning speed, training set size, and the nature of the problem.

Following previous approaches [71], we use the weighted average precision, recall or F-measure (F1) as an evaluation metric. The performance for a particular affect $\overline{P_c}$ is the weighted average performance of that affect over all the subjects: $\overline{P_c} = \sum_{s=1}^{N_s} w_{cs} * p_{cs}$, where $w_{cs} = \frac{N_{cs}}{\sum_{i=1}^{N_s} N_{ci}}$. Here s denotes the index of the subject and c

denotes the index of the class (affect). p_{cs} therefore is the recognition performance on affect c for subject s, and N_s the number of subjects. N_{cs} denotes the number of instances in the ground truth data for subject s that are labeled with affect c. The overall performance $\bar{P}$ can similarly be represented by $\bar{P} = \sum_{c=1}^{N_c} \bar{P}_c / N_c$, where N_c is the number of affects.

Our evaluation is conducted on two spontaneous facial datasets: MSARV and UNBC. Following the same protocol to obtain annotation as in [99][6][92], we define the segment label as pain when OPI≥3, and non-pain if OPI=0. Similarly, for the frame-level label, we follow previous work and determine the label according to the PSPI value [87], where PSPI>0 is labeled as pain, and PSPI=0 is marked as non-pain [6]. This gives us 147 segments and 23 subjects for the UNBC dataset. Likewise, we apply the subjects' self-reported affect as segment annotation and the observer- and self-retrospective judgment as frame annotation. MSARV consists of 11 subjects, each of who has 25 segments.

3.6.1 Evaluation at the segment level

Our first evaluation compares the recognition result at the segment level on both the UNBC and MSARV datasets.

The user-independent model is evaluated using leave-one-subject-out cross-validation. The overall result is the average performance, weighted by the amount of testing data for each individual. To evaluate user-specific learning, we performed leave-one-segment-out cross-validation. Each segment is used for testing in turn, with the remaining segments from the same individual used for training. Similarly, averaging over the test iterations gives us the overall result. We exclude subjects whose segments exhibit only one label type (i.e. only "pain" or only "non-pain") on UNBC, which yielded 22 subjects with 145 segments.

Our state-of-the-art "competitors" are based on two weakly-supervised SVM/MIL-based models from previous work [99][92]. The first model, which we refer to as vMIL (for *vector*-based MIL), uses max-pooling [100], which has been shown to be effective for feature aggregation [99], to extract segment-level features. Each segment is represented by one feature vector, and the individual feature values are chosen as the value of maximum deviation from the mean for that feature. The second model, referred to as sMIL (for *subset*-based MIL), uses the subset representation method, which

represents each segment with cluster centroids from K-Means clustering. We empirically choose K=20 in our experiment. To determine the segment recognition result for sMIL for pain detection on UNBC, we follow previous work [6] and rely on a frame threshold determined by the equal error rate (EER). The classifier for both models is the support vector machine (SVMs) [12], which generally performs well in pattern recognition applications, including state-of-the-art affect detection [71]. Our particular SVMs are implemented by the sequential minimal optimization algorithm [86], using polynomial kernels and parameters determined by grid search. Finally, for affect classification on MSARV, sMIL uses majority voting based on the results of frames in the subset to determine the segment-level result.

Table 3-4 shows that AMIL outperforms MS-MIL [99] and is comparable with RMC-MIL, the current highest-performing approach [92], for user-independent learning. This shows that our information retrieval-based assumption is effective at modeling facial affects.

Table 3-4. Performance and comparison to state-of-the-art MIL methods on user-independent learning, UNBC dataset (Performance metric: accuracy at equal error rate)

MILES [15]	MILIS [33]	MIL-Boost [110]	MI-Forest [61]	MS-MIL[99]	RMC-MIL[92]	AMIL (ours)
78.2	76.9	76.9	75.8	83.7	85.7	84.4

Table 3-5. Result comparison on UNBC and MSARV (Performance metric: accuracy and F-measure)

Dataset	UNBC			MSARV		
Method	vMIL	sMIL	AMIL	vMIL	sMIL	AMIL
User-specific	76.2 (0.74)	76.6 (0.76)	81.6 (0.81)	30.5 (0.33)	53.5 (0.55)	72.0 (0.71)
User-independent	83.7 (0.82)	85.7 (0.86)	84.4 (0.84)	16.4 (0.18)	11.6 (0.13)	32.4 (0.33)

Numbers in and out of the bracket denote the F-measure and accuracy at equal error rate, respectively.

Table 3-5 presents user-specific and user-independent segment-level recognition results on UNBC and MSARV.

For the UNBC dataset, our reconstructed models, vMIL, and sMIL, achieve 83.7% and 85.7% accuracy respectively. This is comparable to reported performance from similar approaches in literature (vMIL and MS-MIL [99] both achieve 83.7%; sMIL and RMC-MIL [92] both achieve 85.7%), and suggests that our reconstructed models are state-of-the-art.

Using AMIL for feature aggregation in user-independent learning achieves performance close to the best result (sMIL: 85.7% vs 84.4% – a difference of 2 segments). When used for user-specific learning in PADMA, AMIL outperforms the other feature aggregation approaches by 5% (81.6% vs 76.6%). This is close to the best performing overall model (user-independent sMIL: 81.6% vs 85.7% – 7 segments). Hence, PADMA achieves performances that are generally comparable to state-of-the-art on UNBC.

Unexpectedly, user-specific learning on UNBC does not perform as well as user-independent learning. Inspecting the data suggests two possible reasons. First, even though UNBC contains a good number of subjects, there is limited data *per subject* (M=1525 frames, SD=712 frames), which makes it difficult for the user-specific model to generalize. Secondly, it appears that the expression of pain may be somewhat more universal, and thus easier to generalize across different users, than other high-level mental states such as interest. This is supported by the fact that pain is usually measured according to the PSPI score, which only considers a small subset of facial action units. In contrast to UNBC, MSARV is relatively richer, with more data per subject and multiple affect labels.

Performance results on MSARV are promising. Table 3-5 shows that the user-specific models significantly outperform their user-independent counterparts across the board. Using AMIL for feature extraction, PADMA achieves the highest performance with 72.0% accuracy and 0.71 F1 – 18% higher than the next best-performing model (sMIL: 53.5% – a difference of 51 segments), and twice as accurate as the best user-independent model (user-independent AMIL: 32.4% – 109 segments). This suggests that, in certain contexts, user-specific models significantly outperform user-independent learning, and the AMIL assumptions provide the best performance.

Data analysis suggests that the performance difference between UNBC and MSARV are mainly due to the difference in their affect attributes. Affects on MSARV include both basic emotions and mental states such as interest and boredom, which commonly occur in daily human-computer interactions. Although it is reported that basic emotions

are universal across cultures [121], in real use, it appears that the *manifestation* of these affects as spontaneous expressions still differ across subjects. It also appears that high-level mental states are manifested differently between people, and may be more challenging to recognize [121]. For instance, some subjects react to boredom by looking away, while others frown or change postures. Modeling this individuality in a user-independent manner would be difficult.

Table 3-6. Confusion matrix and performance of PADMA for segment-level user-specific learning on UNBC. Rows: annotated (truth) class; columns: recognized class. F-Measure over all subjects: 0.81.

	Pain	Non-Pain	Precision	Recall	F1
Pain	34	21	0.85	0.62	0.72
Non-Pain	6	86	0.8	0.93	0.86

Table 3-6 gives the confusion matrix and the performance metrics of PADMA on UNBC. Both precision and recall are high, which demonstrates that AMIL is effective for binary classification.

Table 3-7. Confusion matrix and performance of PADMA for segment-level user-dependent learning on MSARV. Rows: annotated (truth) affect; columns: recognized affect. F-Measure over all affects for all subjects is 0.72.

	N	H	I	B	S	D	F	A	Precision	Recall	F1
N	0	0	1	0	0	0	0	0	NA	0.00	0.00
H	0	27	5	0	0	0	0	0	0.69	0.84	0.76
I	0	6	58	16	4	1	1	1	0.70	0.67	0.68
B	0	4	9	66	2	0	1	0	0.76	0.80	0.78
S	0	0	4	1	7	0	1	0	0.47	0.54	0.50
D	0	1	1	1	0	16	0	1	0.80	0.80	0.80
F	0	1	0	1	1	3	9	1	0.69	0.56	0.62
A	0	0	5	2	1	0	1	15	0.83	0.63	0.71

N: neutral, H: happiness, I: interest, B: boredom, S: sadness, D: disgust, F: fear, A: anger

Table 3-7 shows the same measurements on the multi-class MSARV data. In general, the majority of the segments are recognized correctly. PADMA performs worst on sadness (F1: 0.50), fear (F1: 0.62) and neutral. This is consistent with previous findings

[71], which states that these emotions are naturally more difficult to recognize. The problem is exacerbated in MSARV, since it contains only spontaneous expressions, and most of the time, sadness and fear were not elicited to a high degree.

Post-experiment interviews can help us to understand this phenomenon. For sadness, the subjects noted that they were less likely to feel sad without knowing the context of the video segment. Therefore, if they had previously watched the movie that contains the elicitation video segment, (re)watching the short segment would cause them to recall the movie, and a stronger feeling of sadness is successfully induced. However, if they had not previously watched the movie, they were less likely to feel that emotion. The result is that for some subjects (3 out of 11), the affect of sadness was never successfully aroused during the elicitation process. Fear proved to be another emotion that was hard to elicit. The subjects noted that even though the movie segments might be scary, the fact that they knew that they were in an experiment was counterproductive to eliciting fear.

3.6.2 Evaluation at the frame level

In addition to the segment-level, the frame-level label is also of interest to us, as it is useful for the precise understanding of the temporal affect changes within a segment. For example, it can shed light on the exact moment a patient feels pain, or the moment that a user becomes interested in the stimulus. Frame-level performance can also be considered as an approximation of the gesture-level accuracy.

The frame-level ground truth on UNBC is obtained through the PSPI score, while MSARV provides the frame-level observations.

Table 3-8 presents the frame-level performance on UNBC and MSARV. Unsurprisingly, the frame-level performance is similar to the segment-level performance. User-independent learning with AMIL outperforms user-specific on UNBC, while user-specific learning performs better on MSARV. This may also be a result of insufficient individual training data for user-specific learning on UNBC. More

Table 3-8. Frame-level recognition performance of AMIL on UNBC and MSARV.

Dataset	UNBC	MSARV
User-specific	58.5(0.62)	**59.2(0.59)**
User-independent	**71.3(0.69)**	25.7(0.25)

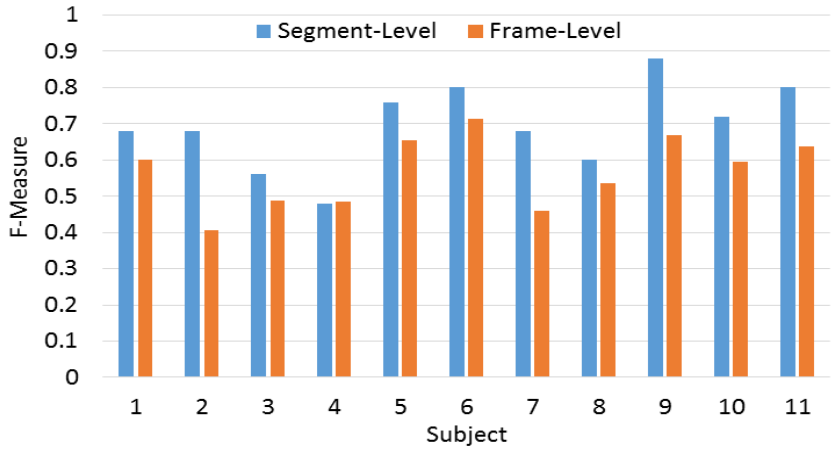


Figure 3-18. Per subject user-specific learning on MSARV. Performance at segment-level and frame-level for all affects.

interestingly, performance at the frame-level is lower than at segment-level in general (71.3% vs. 84.4% on UNBC; 59.2% vs. 72.0% on MSARV).

Figure 3-18 presents the overall F-measure across all affects for each subject on MSARV at the both segment-level and frame-level. For all but one subject, segment-level recognition achieves a higher accuracy. This makes sense as it is challenging to recognize the fine-grained result from the coarse-grained data labels. However, given that we achieve good results on segment-level recognition, this shows that it is possible to extract and label users' facial gestures with only a very small amount of annotation. This also demonstrates that if one wants to obtain the affect implication behind facial gestures, deducing them from the segment-level label would be an effective approach.

The performance difference for different users is also due to a data sparsity problem, as not all affects were successfully aroused in some of the subjects. This suggests that a longer affect elicitation process, or a dynamic affect elicitation process that selects video clips to show the user based on the affects that have already been successfully elicited, might be more effective.

Figure 3-19 shows three examples of facial gestures identified by PADMA. Each image represents one expression label contained in the gesture. This shows that our approach can successfully identify both dynamic and static indicative gestures. For example, gesture (a) represents an expression transition from onset to the apex of happiness. Gesture (b) is associated with boredom by our user, and indicates a fast transition from the onset to the apex and then the offset. Gesture (c) is a static gesture that was associated with sadness. It contains only one expression label, indicating the subject kept her apex expression for a long duration. Sequences (b) and (c) give a sense

(a)

(b)

(c)

Figure 3-19. Examples of identified facial gesture sequences in MSARV. (a) happy, (b) bored, (c) sad. The resulting gestures may contain different numbers of expression labels. Gesture (a) presents a transition from neutral to apex; gesture (c) shows a long-lasting sad expression. Note the subtle difference between (b) and (c).

of the challenge posed by the MSARV data: spontaneous expressions often do not exhibit exaggerated facial muscle movement, which suggests that facial affect detection in real-use situations may be a much more complex task than the posed expression alternative.

For a better understanding, we also investigate the contribution of the PADMA adaptive clustering process, which first chooses a large K and then merges extraneous clusters. We compare our performance against the alternative of using a fixed number of clusters. In our previous experiments, the merging step reduces the number of clusters from 500 to 246~487, depending on the actual facial responses of the subjects. We present experiments with five K values (100, 400, 500, 750, 1000), which lie both

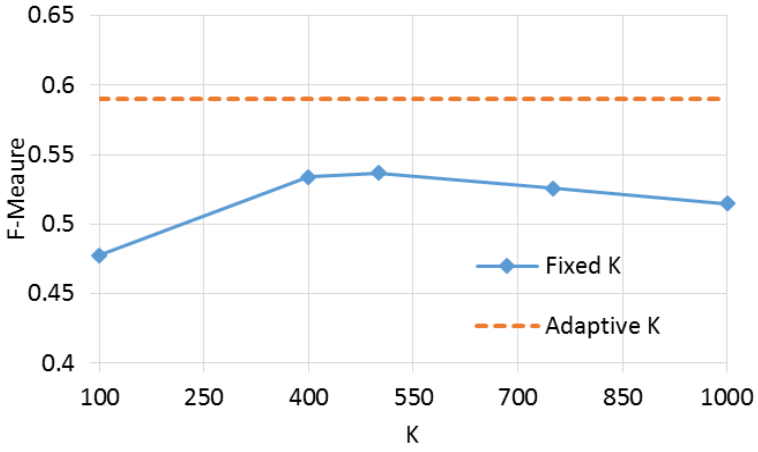


Figure 3-20. Comparison between fixed and adaptive K for PADMA. The adaptive clustering approach clearly outperforms using a fixed number of clusters.

within and beyond the range of the final number of clusters achieved through an adaptive K. It can be seen that the adaptive approach outperforms the fixed approach for all affects (Figure 3-20). This bears out our hypothesis: if K is too small, some of the facial gestures will be merged together, and cause many facial gestures to be labeled as "neutral". In contrast, an over-large K produces different expression labels for similar facial gestures, which makes their distribution sparser and decreases their RFIAF values. We conclude therefore that a proper way to determine the cluster number is essential for facial affect modeling and the post-clustering merging of redundant expression clusters is an essential step in our algorithm.

3.6.3 Learning speed and amount of training data

In this section, we further evaluate the performance of our model as a function of user effort. We have shown that user-independent learning outperforms user-specific learning when there is insufficient training data per user, as in the UNBC dataset. Since time and effort from end-users poses a major challenge for user-specific learning, we wish to understand the impact of training data on system performance.

The evaluation process involves multiple iterations with an incremental training set. User-independent models are evaluated using leave-some-subjects-out cross-validation, and user-specific models with leave-some-segments-out cross-validation. On each iteration, a subset with a certain number of subjects or segments is selected as the training set using a rolling window, with the test set being the rest of the data. The final performance result for each iteration is averaged over all training subsets.

Figure 3-21 shows the impact of training data on performance for the UNBC dataset. The user-specific model starts off with lower performance compared to the user-independent counterpart, but as the number of segments in the training set increases, the user-specific model (best F1: 0.81) rapidly approaches the performance of the user-independent model (best F1: 0.84). Given that the segments on UNBC are generally very short (238 frames per segment on average), this means that the user-specific model can achieve performance comparable to the user-independent model with relatively little effort. However, the learning speed flattens out after 6 segments. This may be due to the fact that UNBC contains on average only 6 segments for any individual subject. Though the user-specific model does not outperform the user-independent model, the performance difference is small; and the steep upward trend of the learning curve

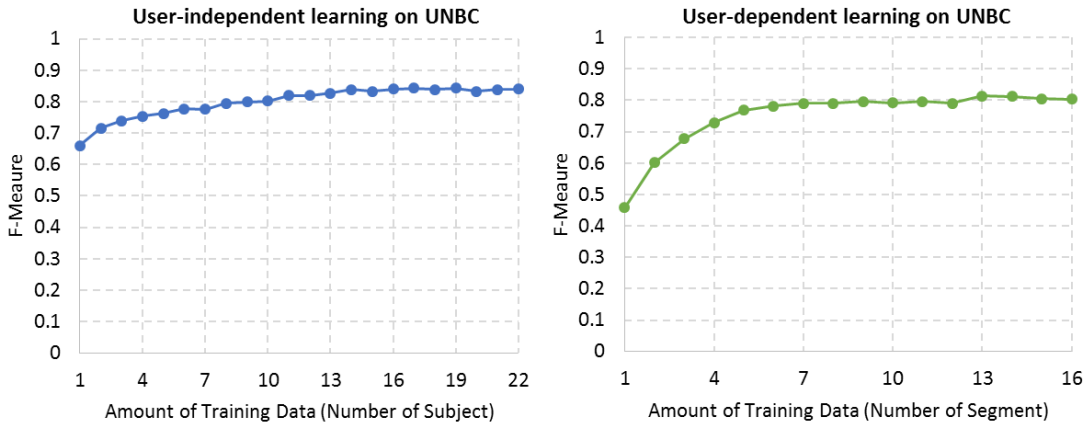


Figure 3-21. Learning speed vs amount of training data on UNBC. Comparison between user-independent and user-specific learning of AMIL. The user-specific model shows a faster learning speed.

suggests that given enough training data, it is likely that the user-specific model would outperform the user-independent model.

To validate this hypothesis, we run similar experiments on the MSARV dataset, where more data per subject is available. We use the boredom affect as an example to investigate the amount of additional effort required to extend an existing model to incorporate additional affects. On the user-specific model, we train on all response video segments self-reported as *not* boredom and increment the amount of training data by adding one boredom-labeled segment (≈1min per segment) at a time, each time testing on the remainder of the boredom-labeled segments. Likewise, for the user-independent model, we train a basic model on a subset of subjects and perform leave-some-subjects-out cross-validation by incrementing the amount of training data one subject at a time (≈6.8min of new boredom-labeled data per subject), while testing on the rest of the subjects.

Figure 3-22 compares the learning speed between the user-specific and user-independent models, illustrating performance as a function of the time required from subjects. The points on the curve represent the F1 performance for the "boredom" affect for that iteration.

The results are encouraging. The learning speed of the user-specific model increases much faster than the user-independent model. Five more training segments (≈6min) increase F1 of the user-specific model by 0.3. For the user-independent model, however, adding data from 9 more subjects (≈60min) improves F1 by only 0.22. In

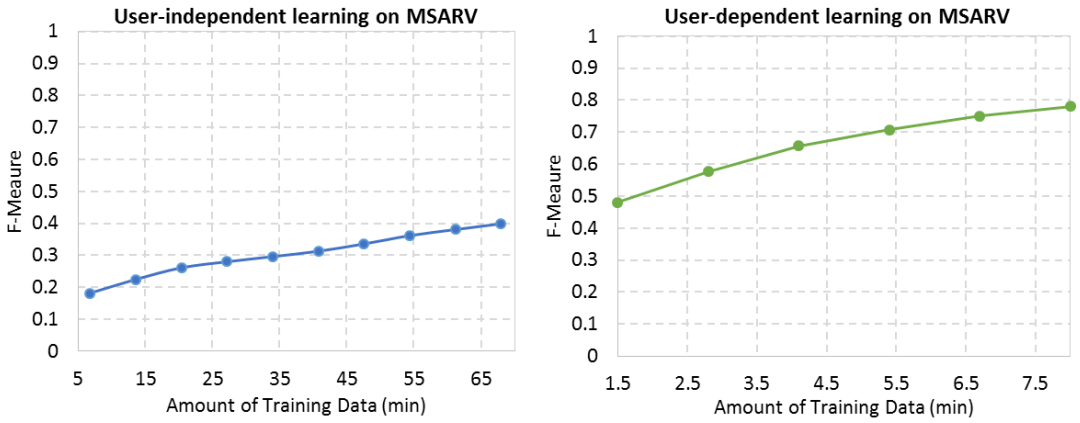


Figure 3-22. Learning speed vs amount of training data on MSARV for the "boredom" affect. Comparison between user-independent and user-specific learning of AMIL. Even with small amounts of individual data, the user-specific model outperforms the user-independent model in both learning speed and accuracy.

addition, the user-specific model, trained on one segment, already outperforms the user-independent model trained on data from 10 different subjects.

We conclude that given sufficient weakly-labeled individual data, user-specific learning can outperform user-independent learning. Furthermore, to achieve comparable performance, the training set required for the user-specific technique is relatively small compared to that of the user-independent techniques.

3.7 Summary

This chapter presents the AMIL technique to measure the facial gesture indicativeness and provides the relevant facial behavior analysis on the MSARV and UNBC datasets. The data reveals the general consistency with a certain degree of variation between the overall human perception and fine-grained facial behaviors, i.e. between the segment- and frame- level annotations. Additionally, our data also supports the effectiveness of appraising the gesture-affect association through RFIAF.

It also presents PADMA, a method of building user-specific models for facial affect recognition, which uses novel algorithms for adaptive clustering and association-based multiple-instance learning. Experiments demonstrate that PADMA can effectively extract a user's facial gestures and correctly assign the corresponding affect labels without the need for human annotation at the frame level.

To verify the efficacy of our approach, we present evaluations comparing our method with related weakly supervised models on both user-specific and user-

independent learnings. Our results conclude that PADMA can successfully model spontaneous facial affects in a practical manner.

Our experiments demonstrate the effectiveness of PADMA on the UNBC and MSARV datasets. It is not difficult to see that this approach can be directly applied to everyday computing activities. For instance, a user could update the model by self-reporting his/her feelings every time after watching a YouTube video that he/she feels strongly about. This should result in a higher accuracy than by trying to elicit diverse emotions within a short time period. Moreover, continuous data collection in real-use situations will provide more comprehensive expression data and more accurate gesture distribution models, which will further improve the generalizability and accuracy of the model.

Chapter 4

Fast-PADMA – Going from User-Specific to User-Adaptive

Selected notations used in this chapter

a_i the *i-th* affect
$\boldsymbol{C}_i$ the clustered set of D_i
$\boldsymbol{c}_i$ the *i-th* facial centroid feature vector
D_u^s segment data of the *u-th* source subjects
D^t segment data of the target subject
h_i the *i-th* weak generic classifier
G_t the set of gestures occurring in the windows that span over the *t-th* element in the run-length encoded sequence
g_j the *j-th* facial gesture
N number of source subjects
n^t number of available segment of the target subject
$n^{(u)}$ segment number of the *u-th* source subject
$n_j^{(u)}$ instance number of the *u-th* source subject's the *j-th* segment
$\boldsymbol{M}$ jitter-similarity matrix for cluster merging
m_{pq} the element in the *p-th* row and the *q-th* column of $\boldsymbol{M}$
$\boldsymbol{w}$ weighting vector of the weak generic classifiers
S a set of statistics descriptors
s+1 moving window size for segment-affect association vector calculation, =5
t_{pq} transition frequency between centroid *p* and *q* in the sequences of centroid IDs
V the set of all response clip-sets
$|V|$ the number of different self-reported affects
v_i the response clip-set for affect a_i, a group of video segment labeled with the same affect
$\boldsymbol{X}_j^s, \boldsymbol{X}_j^t$ the *j-th* segment data of a source and target subject, respectively
$\boldsymbol{x}_i^s, \boldsymbol{x}_i^t$ the *i-th* instance in a segment of the source and target subject, respectively
$\boldsymbol{x}^{(n)}$ neutral vector representing the expressionless facial expression
$\hat{\boldsymbol{x}}$ transformed feature vector to minimize the personal differences
y_i^s, y_i^t the *i-th* class label of the source and target subject
$\boldsymbol{z}$ segment-level feature vector
$\boldsymbol{z}_a$ segment-affect association vector of the overall gestures contained in a segment
$\hat{\boldsymbol{z}}_a$ normalized segment-affect association vector
$\boldsymbol{z}_s$ pooling feature vector of a statistics descriptor
λ expectation of within-cluster distances
σ_c standard deviation of pairwise centroid distances
σ_p standard deviation of transition frequency from centroid p to others
ϕ_s pooling operation of different statistical descriptors
ε classification error of a weak generic classifier on D^t

Generally speaking, a generic classifier is susceptible to the personal geometric bias and the emotional expression difference. In contrast, a user-specific model may be able to fully adapt to these individualities. However, the performance of a user-specific model is commonly constrained by the amount of well-annotated individual data. A promising solution can be a hybrid model learning the commonality from the source data and accommodating the individual distinctiveness based on the target data.

On the basis of the user-specific model discussed in Chapter 3, this chapter intends to extend the user-specific model with a user-adaptive approach. Instead of adopting only the target user's data, a user-adaptive model also learns from dissimilar source subjects. The proposed method relies on proper alignment of data from different individuals and an efficient ensemble mechanism to classify pain, arousal, and valence across four publicly available spontaneous facial datasets with promising performance.

This chapter begins with an overview of the proposed method. Then the segment-level feature representation will be introduced, followed by an explanation of the supporting techniques for the practical user-adaptive model. We then present the experimental results on different facial datasets and conclude with a discussion of the validity and effectiveness of the proposed method.

4.1 Designing the framework for User-Adaptive Facial Affect Modeling

The objective behind this approach is to rapidly learn a spontaneous facial affect recognizer for practical use. Our motivation is the fact that building an accurate user-specific model requires a large amount of individual data. Although the MIL techniques significantly reduce the annotation effort by requiring only one overall label per video segment, collecting sufficient individual data (multiple well-annotated segments) is still time-consuming. We, therefore, attempt to accelerate the personal model learning by exploring the source knowledge from other training subjects. In addition, to ensure the application feasibility in real use, we avoid the computationally expensive optimization that is usually used in knowledge transfer techniques.

The proposed affect recognizer is trained on video segments of different subjects. Only one segment-level annotation of self-reported affect is required per segment. As shown in Figure 4-1, geometric facial features are automatically extracted frame-by-frame from the video segments. The association between facial gestures and affects is

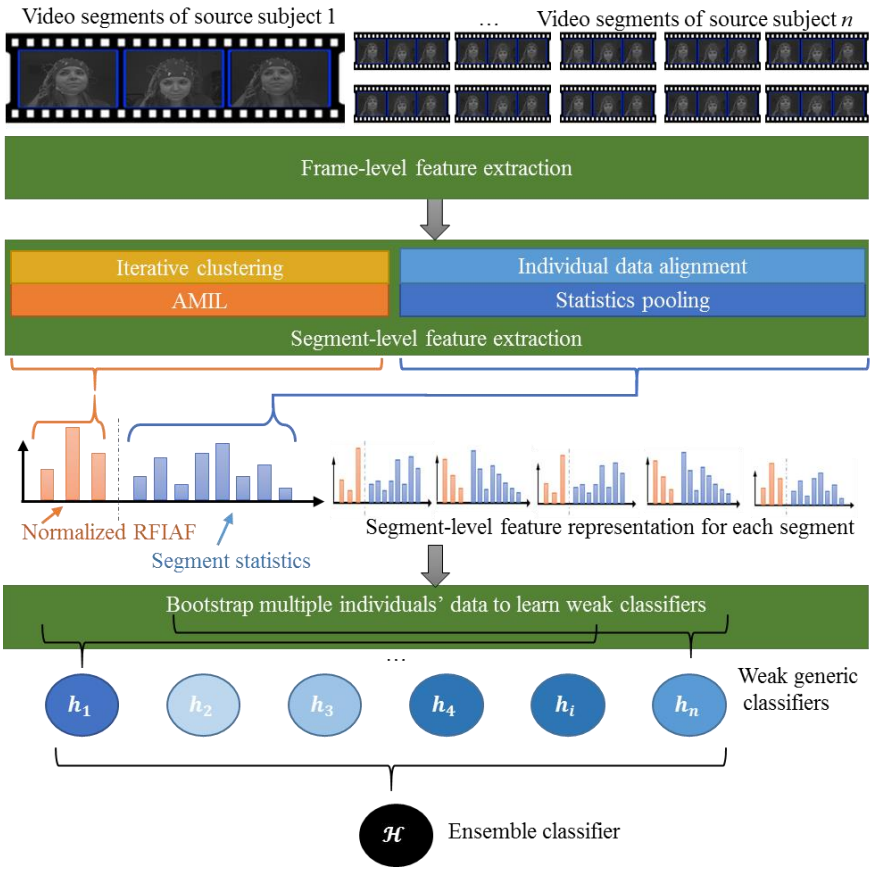


Figure 4-1. System overview of the proposed method. Geometric facial features are automatically extracted from video segments. Segment-level feature representation is obtained by AMIL and pooling. Individual data alignment is adopted to mitigate the personal geometric bias. Ensemble classifier is then learned from the bootstrapped data from the source dataset.

acquired by AMIL [46]. However, the result of AMIL represents only the overall association between facial gestures and affects in a segment, and it is extracted from a user-specific aspect. In order to preserve the instance distribution characteristic and make use of the valid generic knowledge, fast-PADMA combines the AMIL result with the descriptive statistics, such as the standard deviation and kurtosis of the temporal sequence of each facial feature, as the final feature representation for a segment.

To alleviate the individual geometric bias, we suggest a personal data alignment technique to handle the facial change that takes into account the "expressionless" states of the individual. We construct multiple *weak generic classifiers*, each of which is trained on *a subset of* the source subjects' data (our evaluation also includes the comparison with the performance of *weak individual models*, each of which is trained on *one* source subject's data). Fast-PADMA adapts to a particular target user by adjusting the weights of each weak generic classifiers. During the application phase,

the available target data is used to evaluate and re-weigh the weak generic classifiers. As illustrated in Figure 4-1 the weighting (indicated by blue intensity) of each weak generic classifier can be different, according to the actual data of the target user. The final classification result depends on the weighted results of all of the weak generic classifiers.

4.2 Modeling the User-Adaptive Facial Model

We assume there are *N* source subjects in the training set, $D^s = \{D_u^s\}_{u=1}^N$. Data from the *u-th* source subject is denoted as $D_u^s = \{\boldsymbol{X}_j^s, y_j^s\}_{j=1}^{n^{(u)}}$, where $n^{(u)}$ is the number of video segment. $\boldsymbol{X}_j^s = \{\boldsymbol{x}_i^s\}_{i=1}^{n_j^{(u)}}$ is the set of instances in the *j-th* segment, where $n_j^{(u)}$is the number of instances in this segment. For a particular target user, we also know $D^t = \{\boldsymbol{X}_j^t, y_j^t\}_{j=1}^{n^t}$, n^t denotes the number of available segment in D^t. Each instance in our case is a 20-dimension frame-level facial feature vector, i.e. $\boldsymbol{x}_i^s, \boldsymbol{x}_i^t \in \boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{X}} \in \Re^{20}$ is the frame-level feature space; $\boldsymbol{z} \in \boldsymbol{\mathcal{Z}}$ is the segment-level feature vector and $\boldsymbol{\mathcal{Z}}$ the segment feature space; $y_j^s, y_j^t \in \mathcal{Y}$ represents the class labels. Our objective is to build an adaptive classifier to identify the label for a set of unseen instances of the target user $\boldsymbol{X}^t$ based on $D = D^s \cup D^t$, i.e. $f_T: \boldsymbol{X}^t \longrightarrow y^t$.

4.2.1 Frame-level facial behaviors extraction

Although most recent facial expression recognition systems tend to prefer complicated appearance-based features such as SIFT and LBP, fast-PADMA makes use of easily computed geometric features. Geometric features are simple and can be acquired with low computation, which is reasonable for real-use facial affect systems. More importantly, these features can be easily interpreted as they reflect the facial change in a straightforward manner. This intuitiveness facilitates the determination of the segment-level representation feature, since it ensures the corresponding values of statistics features are also interpretable.

Following the method presented in [46] to extract geometric facial feature, we apply the Supervised Descent Method [118] to automatically track the 66 facial landmarks from video segments. The same as the procedure described in the last chapter, a 3D landmark model is registered to fit the 2D landmarks' projection in the 3D space. The

geometric facial features are then calculated in the projected space to as given in Table 3-2.

4.2.2 Segment-level features aggregation

We construct a segment-level feature representation, which includes the overall implication of facial gestures and the pooled statistics of frame-level features. AMIL is used to calculate the overall affect association of facial gestures in each segment and the regular descriptive statistics to reflect the instances distribution.

Extracting facial gestures

We follow the practice of [46] to identify facial gestures from the facial response, which is encoded by the facial cluster centroids. An iterative k-means clustering is used. The number of clusters is determined by thresholding the intra-cluster instance similarity, approximated as a function of λ. The value of λ (=.25 for the normalized features) is empirically determined, according to the appearance of the resulting centroids. We therefore select a value of k_i that partitions the source individual data D_i into k_i sets, such that each set satisfies

$$\operatorname{argmin}_{\boldsymbol{C}_i} \sum_{j=1}^{k_i} \sum_{\boldsymbol{x} \in C_{ij}} \left\| \boldsymbol{x} - \boldsymbol{\mu}_j \right\| \tag{4.1}$$

$$\text{s.t.} \, \forall j \, \left| \left\| \boldsymbol{x} - \boldsymbol{\mu}_j \right\| - \lambda \right| < \varepsilon$$

where $\boldsymbol{C}_i = \{C_{i1}, \ldots, C_{ik_i}\}$ is the partitioned set of D_i; ε denotes a small value. By replacing the instances with their cluster centroids, the sequential instances in each segment can be converted into the sequence of centroid IDs.

Due to the natural visual noise in the video stream, the localization of the facial landmarks can be noisy, which results in the temporal jitter of the geometric features. If the facial expression frequently bounces back and forth between similar centroids in the centroid ID sequences, we merge the two clusters. Rather than iteratively taking turns to consider the centroid similarity and jittering frequency as depicted in Table 3-3, we use a sophisticated method to simultaneously cope with these factors. We construct the jitter-similarity matrix $\boldsymbol{M}$ and apply the normalized cuts [51] to identify the clusters to be merged. Each element m_{pq} represents the loss of merging of cluster p and q:

$$m_{pq} = \exp\left(-\left|\frac{\left\| \boldsymbol{c}_p - \boldsymbol{c}_q \right\|}{\sigma_c}\right|^2 + \left|\frac{t_{pq}}{\max\left(\sigma_p, \sigma_q\right)}\right|^2\right) \tag{4.2}$$

where $\boldsymbol{c}_p, \boldsymbol{c}_q$ indicate the feature vectors of two centroids, σ_c the standard deviation of pairwise centroid distances, t_{pq} the corresponding transition frequency, and σ_p the standard deviation of transition frequency from centroid p to others. Facial gestures are then identified from the temporal moving window across each segment as described in last chapter.

Identifying segment-affect association

In contrast to the conventional MIL methods that select some instance(s) as the prototype to represent a bag, AMIL explores all the potential indicative instances. It assumes that if an instance occurs frequently in segment(s) labeled with one particular class, but not in others, this instance shares a strong association with that class.

Based on the run-length encoded sequences of centroid IDs, we calculate the RFIAF value to reflect the affect implication of a gesture g_j. Given the values between each gesture and affect, we can then construct a vector $\boldsymbol{z}_a$to represent the segment-affect association of the overall gestures contained in a segment. Each of its element z_{ai} indicates the segment association with a particular affect a_i,

$$\mathrm{z}_{ai} = \sum_{t=s/2}^{n-s/2} \sum_{g_j \in G_t} \mathrm{RFIAF}(g_j, v_i, V) \tag{4.3}$$

where G_t indicates the set of gestures occurring in the moving window spanning over s+1 frames with its center at the *t-th* frame of a segment; n is the number of frame in a segment; v_i the clip-set labeled as affect a_i; and V the overall clip-set. We then use the normalized $\boldsymbol{z}_a$ as partial of the segment-level features to represent the overall degree of segment-affect association:

$$\hat{\boldsymbol{z}}_a = \frac{\boldsymbol{z}_a}{|\boldsymbol{z}_a|} \tag{4.4}$$

Pooling instance statistic

Although $\hat{\boldsymbol{z}}_a$ can reflect the essential gesture-affect associations of the overall segment, it does not reflect the inner characteristics of the segment, such as the instance dispersion and the distribution attributes. We believe that such characteristics offer the structural cues of the instances inside a segment and can be informative segment representation.

Pooling is a well-used technique to aggregate frame-level feature to segment-level representation [46][99]. In order to provide a further description to adequately represent a segment, we adopt the pooling technique to derive statistical information from

segments. Denoting ϕ_s as the pooling operation of different statistical descriptors, we define

$$\boldsymbol{z}_s = \phi_s(\boldsymbol{x}_j | \boldsymbol{x}_j \in \boldsymbol{X}) \tag{4.5}$$

as the resulting feature vector of the statistic pooling operation. The suggested descriptors ϕ_s can include the dispersion attributes such as mean, median, standard deviation, variance, minimum and maximum, and the distribution attributes such as skewness and kurtosis.

Since the frame-level feature representation is in a 20-dimension space $\boldsymbol{\mathcal{X}} \in \Re^{20}$, concatenating the segment statistics features $\boldsymbol{z}_s$ of one descriptor ϕ_s leads to an increment of 20 dimensions for the final segment vector $\boldsymbol{z}$. Our pilot experiments, however, confirm that the pooled features of mean and median contribute similarly to the final recognition, as do those of standard deviation and variance. We therefore exclude the descriptors of median and variance to reduce vector dimensionality in our experimental evaluation.

The segment feature representation can be shown as

$$\boldsymbol{z} = \langle \hat{\boldsymbol{z}}_a, \boldsymbol{z}_{s_i} | s_i \in S \rangle \tag{4.6}$$

where s_i indicates a particular statistics descriptor in the set of descriptors S, which include mean, standard deviation, minimum, maximum, skewness and kurtosis. In other words, it is a concatenated vector of the normalized segment-affect association vector $\hat{\boldsymbol{z}}_a$ and the pooled statistics vectors.

Individual Data Alignment

Spontaneous expressions are often subtle in nature. Visually, the difference that results from facial expression changes is usually marginal, compared with the difference resulting from personal appearances. This means that when it comes to learning spontaneous facial affect from different source subjects' data, it is critical to perform individual data alignment to emphasize the emotion-induced facial deformation, and reduce identity bias.

Our study focuses on the scenario where limited data of target user is available. Utilizing the labeled target data as well as the source subjects' data allows us to identify the personal attributes, and therefore, transform and align the individual data to a normalized feature space. This will alleviate that the impact of the personal geometric bias on the facial affect recognition process.

We make the assumption that a similar physical movement, as reflected in the deformation of the geometric facial features, is caused by a similar implication of affect across subjects. We then use unity-based normalization to align feature boundaries (minimum and maximum) of different subjects to the same points in the normalized feature space. Data of different individuals in the normalized feature space hence have the same boundaries. In other words, we assume, for example, the maximum observed degree of mouth openness of different subjects indicates the same affect to a similar degree, such as equal intensity of surprise. A similar process has been used by Soleymani et al. [101] to reduce the differences among participants.

Apart from the feature boundary alignment, previous studies also show that normalizing with respect to the expressionless or the neutral frame is vital in removing subject-dependent bias [99][71]. A conventional method is to calculate the landmark displacement relative to the neutral frame. For instance, displacement features are obtained by subtracting the x and y coordinates of the facial landmarks in each frame from the corresponding coordinates in the first frame (neutral frame) in CK+ [71]. The rationale for this normalization is to construct a displacement feature space, where features share similar indicativeness to affect. This feature normalization, however, contains two main drawbacks. First, without accounting for the individual feature boundaries, it assumes the displacement implication is identical across subjects. This assumption, however, is not valid due to the diversity of individualities. Second, it requires identifying the neutral frame. In datasets such as CK+, this is the first frame of each sequence. However, this is not always the case in real-use situations, and most other spontaneous facial datasets do not follow this protocol.

(a) Identifying a neutral face

We, therefore, propose an automated approach to identifying the neutral frame from the weakly annotated data. A naïve assumption is to assume that the neutral expression is the most frequent centroid in the centroid ID sequences. However, this is not true for most of the video-elicited datasets. Consider, for instance, a good number of amusing elicitation videos may lead to a majority of smiling faces in the dataset. Others may show well-activated expressions towards the stimuli while the neutral expression does not show frequently. Furthermore, depending on the success of the emotion elicitation, the lengths of the stimuli also influence the expression distribution.

To identify the neutral expression in a proper manner, we assume neutral *as the expression that occurs most frequently across video clip-sets with different self-*

reported affect labels. It is not difficult to see that this accords with the indicativeness of a facial gestures, as defined through the RFIAF measure from previous work [46], which takes into account both the prevalence of a gesture and its rarity over all affects. Given the centroid ID sequences, we, therefore, introduce the RFAF to measure the *non-indicativeness* of an expression. The RFAF of an expression c_j is defined as:

$$\mathrm{RFAF}(c_j, V) = \sum_{v_i \in V} \mathrm{RF}(c_j, v_i) * \mathrm{AF}(c_j, V) \tag{4.7}$$

where AF measures generality of c_j among different affects:

$$\mathrm{AF}(c_j, V) = log \frac{\left|\{v \in V : f(c_j, v) > 0\}\right|}{1 + |V|} \tag{4.8}$$

By selecting the facial centroid with the maximum RFAF value to represent the neutral frame $\boldsymbol{x}^{(n)}$:

$$\boldsymbol{x}^{(n)} = \mathrm{argmax}_{c_j} \mathrm{RFAF}(c_j, V) \tag{4.9}$$

we then have $\boldsymbol{x}^{(n)} = \langle x_1^{(n)}, \ldots, x_{20}^{(n)} \rangle$, where each $x_i^{(n)}$ denotes the value of a geometric facial feature in the neutral frame of a particular subject.

(b) Aligning data according to neutral and boundary values

Taking into consideration both the neutral and the boundary values of each feature, we define a mapping function to align the geometric features,

$$\hat{\boldsymbol{x}} = \phi_a(\boldsymbol{x}) \tag{4.10}$$

In ϕ_a each feature of the individual data is normalized by a piecewise function for the simplicity, leaving the complex non-linear transformation to be learnt by the supervised classifier. For the *i-th* geometric facial feature x_i, we align it to

$$x_i' = q_0(x_i)^{\theta} q_1(x_i)^{1-\theta} \tag{4.11}$$

where θ is a Heaviside step function:

$$\theta = \frac{1}{2}(1 + \mathrm{sign}(x_i - x_i^{(n)})) \tag{4.12}$$

$q_0(\cdot)$ and $q_1(\cdot)$ are the scaling functions:

$$\begin{aligned} q_0(x_i) &= \frac{1}{2} \cdot \frac{x_i - x_i^{(n)}}{x_{i,max} - x_i^{(n)}} + \frac{1}{2} \\ q_1(x_i) &= \frac{1}{2} \cdot \frac{x_i - x_{i,min}}{x_i^{(n)} - x_{i,min}} \end{aligned} \tag{4.13}$$

where $x_{i,min}$ and $x_{i,max}$ represent the minimum and maximum values of *i-th* geometric facial feature across all segments of a particular subject.

After individual data alignment, (6.5) and (6.6) can be rewritten as:

$$\hat{\boldsymbol{z}}_s = \phi_s(\phi_a(\boldsymbol{x}_j)|\boldsymbol{x}_j \in \boldsymbol{X}) \tag{4.14}$$

and

$$\boldsymbol{z} = \langle \hat{\boldsymbol{z}}_a, \hat{\boldsymbol{z}}_{s_i} | s_i \in S \rangle \tag{4.15}$$

Figure 4-2 illustrates the alignment result of two individuals. The purple axes in (a) and (c) indicate the 2D projection of the data in the raw feature space before alignment. Each inner curve represents a level of data density. Red and yellow dots denote the boundary values and the point of neutral, respectively. As shown in the figures, the neutral point is not necessarily the densest point in the distribution. Aligning the individual data independently according to Equation (4.10) transforms the distributions to the normalized feature space as shown in Figure 4-2 (b) and (d). The general shapes of the transformed data distribution maintain highly consistent with the raw distributions, however, the boundaries are aligned to the same values across subjects. Two distributions fall in the same bounding box in Figure 4-2 (e). And the neutral points of different subjects are aligned to the center of the normalized space.

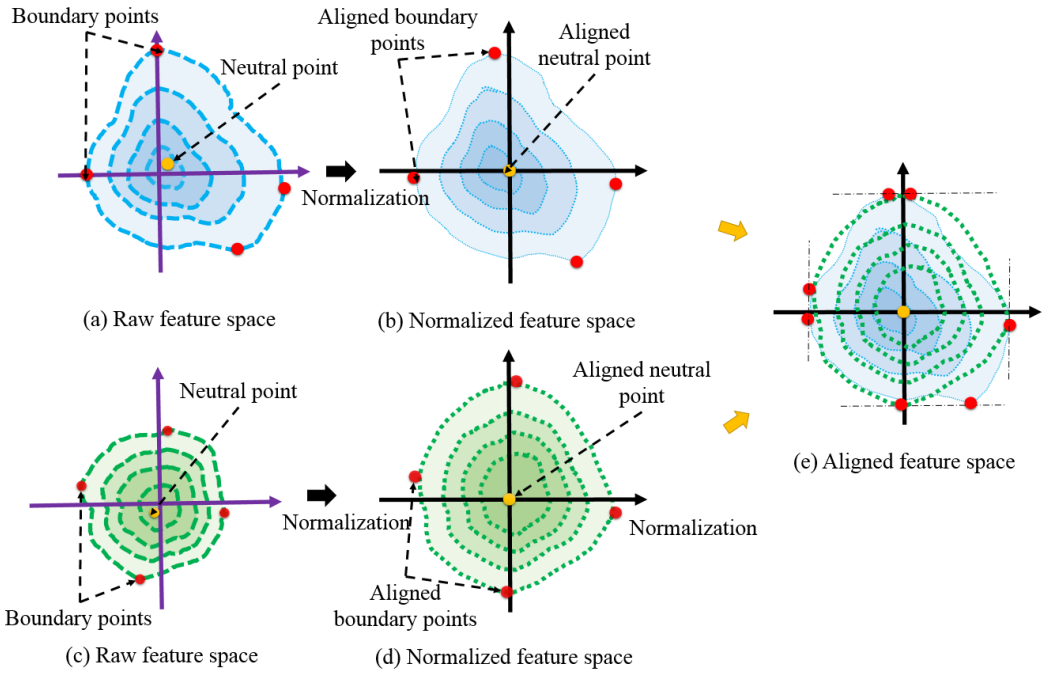


Figure 4-2. Illustration of data alignment of two subjects. The purple axes in (a) and (c) indicate the 2D projection of the data in the raw feature space before alignment. Each inner curve represents a level of data density. Red and yellow dots denote the boundary values and the point of neutral, respectively. (b) and (d) show the transformed distributions to the normalized feature space of (a) and (f), respectively. Putting (b) and (d) on the same axes (e), it can be seen that the boundaries of the two subjects are aligned with the same values and their neutral points are aligned to the same point as well, in the center of the normalized feature space.

Compared with previous work, in particular, the unity-based normalization in [101], which uses two points for alignment (min & max), our method applies one more fiducial point, i.e. the neutral point, to align the distributions of different individuals. In this sense, we explicitly take into account the geometry of the neutral facial expression for our classifiers.

4.3 Building the Ensemble Classifier

In this section, we describe a bootstrapping strategy to build an ensemble classifier for spontaneous facial affect recognition. Unlike the conventional generic classifier that is trained on all the source subjects' data, D^s, we learn a set of classifiers, each of which is trained on a subset of D^s. More specifically, given N source subjects, we train on different $N-1$ subjects' data to learn one weak generic classifier each time. Consequently, we are less likely to suffer from the overfitting issue of building a weak classifier on the limited data of individual source subject.

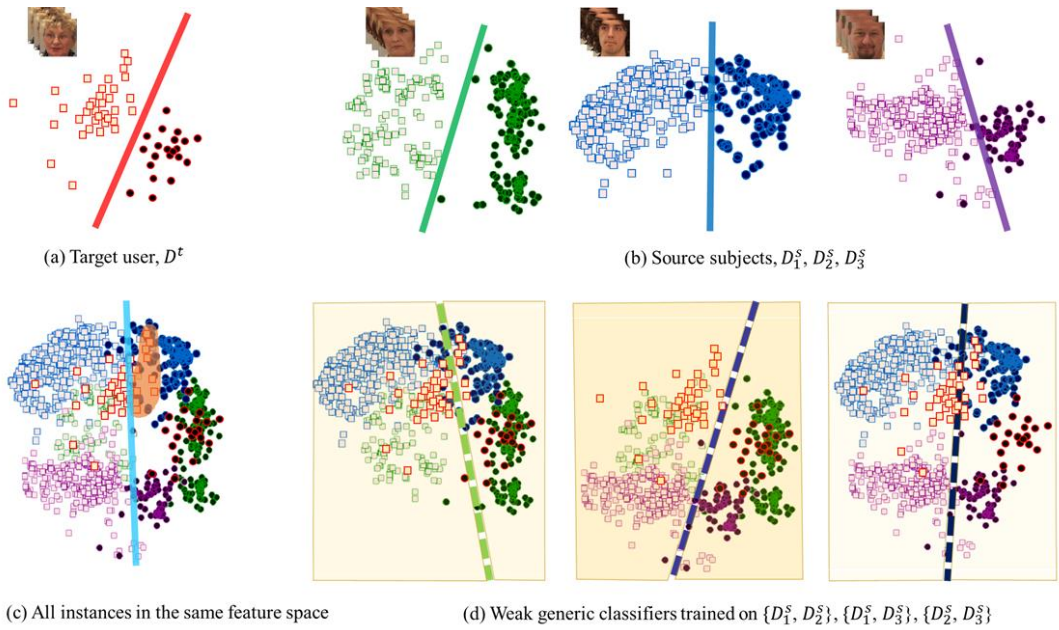


Figure 4-3. Illustration of the data projections in the proposed method. (a) and (b) present the data distribution of the target user and source subjects, respectively. Positive and negative instances are indicated by the squares and circles. The solid lines in (a) and (b) denote the ideal hyperplanes of each user-specific model. (c) shows the projection of the source and target data in the same feature space. The orange shadow points out the conflicting instances between D^t and D_2^s, which will be misclassified by the generic classifier learnt directly from all the training instances. (d) demonstrates the weak classifiers learnt on the bootstrapped data, each of which excludes one different source subject. The dash lines indicate the hyperplanes of the weak generic classifiers. The background transparency denotes the corresponding weight of the weak generic classifier, which depends on the performance on the available target set.

Our hypothesis is that there are some conflicting instances between the target and the source subjects that locate closely in the feature space, but are annotated with contradictory affects. Figure 4-3 (a) and (b) show an illustration of the 2D data projection of the target user and three source subjects. Although positive and negative instances of the target user appear to be nicely separable in (a), projecting all these data to one space (c) shows that some of the target instances (in the orange region) have conflicting annotations with instances in D_2^s, which means the hyperplane learnt from the generic data can be problematic. Figure 4-3 (d) indicates three classifiers trained on the different data combinations of the source subject. It is intuitive that removing the source subject that contains the conflicting instance leads to a clear discrimination of the target data. However, the conflicting instances are generally spread across different source subjects. It is impractical to identify all these instances in advance. It also does not make sense to throw out all data from that source subject, as the occurrence of conflicting instances do not rule out the contribution of other instances from the same source subject. Rather than completely discard a data subset, we retain all the weak generic classifiers and weigh them based on their performances on the available target set. As shown by the background intensity of Figure 4-3 (d), weak generic classifiers performing well on the available target set are given a high weight.

Given a source set D^s and a target set D^t, we propose to learn an ensemble classifier for the target user and infer the label y^t for an unseen target set $\boldsymbol{X}^t$. The algorithm is summarized in Table 4-1. We extract $\boldsymbol{z}^t$ as the segment representation of $\boldsymbol{X}^t$ and predict the label y^t according to a weighted voting mechanism:

$$\text{argmax}_{y^t} \sum_{n=1}^{N} w_n I(h_n(\boldsymbol{z}^t) = y^t) \tag{4.16}$$

where $I(\cdot)$ is an indicator function and w_n is the weight for a weak generic classifier. In our experiments, we use SMO [56] to build the weak classifiers, with the cost parameter set to 0.05.

In addition to reducing the impacts of the conflicting instances and alleviating the overfitting issue, our proposed method has advantages over the transfer technique in its simplicity and efficiency. To apply this method, we only need to evaluate the performance of the weak generic classifiers on the available target set. No computational distribution fitting or similarity calculation is required. For the same reason, the huge source dataset does not need to be deployed, stored or used for

Table 4-1. Learning the ensemble classifier.

Input: source set D^s and the training set of the target subject D^t
Output: A set of weak generic classifiers $\mathcal{H}$ for the target user and the corresponding weights $\boldsymbol{w}$
Phase-I Learning weak classifier set $\mathcal{H} = \{h_i\}$ **from** $D^s = D_1^s, \ldots, D_N^s$
for n = 1 to N
Resample a subset of training data from source set, $D_n = D^s \backslash D_n^s$
Learn a weak generic classifier h_n on D_n
Update the weak classifier set $\mathcal{H} \leftarrow \mathcal{H} \cup h_n$
end for
Phase-II Learning an ensemble classifier on target training set D^t
for n = 1 to N
Evaluate h_n on D^t, classification error $\varepsilon = \sum_{j=1}^{n^t} y_j^t \neq h_n(\boldsymbol{z}_j^t)/n^t$
Update weights $w_n = (1-\varepsilon)/\varepsilon$, $\boldsymbol{w} \leftarrow \boldsymbol{w} \cup w_n$
end for
return $\{\mathcal{H}, \boldsymbol{w}\}$

retraining. This is essential for the rapid building of the target facial affect model in real-use situations.

4.4 Experimental Evaluation

Given a new user, fast-PADMA (1) aligns the target user data with data from our source subjects using the neutral point and the boundary values; (2) extracts and segment-level feature representation; and (3) constructs an ensemble classifier from data of source subjects who are similar to the target user. Our evaluation therefore present substantiated experimental results for these 3 aspects and show the effectiveness of fast-PADMA by comparing against user-specific, generic and hybrid models.

For fairness in evaluation, all models use the same machine learning classifier (SMO) and the features are kept constant. However, AMIL relies on the availability of a bag label to obtain the associations between expressions and affects. Since the generic model would not have access to target user data, segment features extracted by AMIL from the target data were excluded from this model.

The performance of models trained using target user data is measured via leave-one-segment-out cross-validation and the generic classifier trained without target data is evaluated via leave-one-subject-out cross-validation. The reported result is the weighted average across segments and subjects.

4.4.1 Evaluation data

We evaluate our method on four datasets: binary pain/non-pain on the UNBC dataset, and 3-class arousal and valence on DISFA, MSARV, and MAHNOB. Before diving into the evaluations, we provide brief introductions of these datasets:

The UNBC dataset contains 200 segments of near frontal facial expressions from 25 subjects with potential shoulder pain. The length of each segment ranges from 58 to 518 frames. Expert coders used Observer Pain Intensity (OPI) rating to score the segments on a 6-point scale (0: no pain; 5: strong pain). Following the protocol of previous studies [99][6][92], we re-define the binary segment-level label with OPI≥3 as "pain" and OPI=0 as "no pain" and omit the segments with intermediate intensity. Excluding subjects with only one data segment gives us 23 subjects and 147 segments.

The DISFA dataset consists of 27 subjects. Each subject has 9 frontal spontaneous response segments to a set of emotional stimuli. The total length of the stimuli is around 4 minutes. The original annotation of DISFA is the frame-level activation level of the facial AUs. To acquire the segment-level affect annotation, we recruited 6 postgraduate students aged 21-31 (3 female) to manually identify the affects appearing in each segment, including anger, disgust, fear, happiness, sadness, surprise and neutral if none of above appears. The annotated results across different observers show a high degree of agreement with an average Fleiss' Kappa of 0.606 across all emotion categories.

MSARV is composed of 11 subjects' frontal facial response to emotion stimuli, which were collected with the front camera of a mobile phone. Compared to DISFA, each subject has a large amount of data (25 segments). The total length of the stimuli is around 41 minutes. The original segment annotation includes neutral, happiness, interest, boredom, sadness, disgust, fear and anger. As we focus our study on arousal and valence classification in this chapter and there is no clear mapping from boredom and interest into arousal and valence, we therefore exclude the data that only has one of with these two labels. This gives us 11 subjects and 106 remaining segments.

MAHNOB has two distinct paradigms, the emotion experiment (affect classification) and the implicit tagging (agreement/disagreement classification). We

Table 4-2. Mapping emotion into three classes on arousal and valence [101]

Arousal classes	Emotion
Calm/Low arousal	Sadness, disgust, neutral
Medium arousal	Joy and happiness, amusement
Excited/High arousal	Surprise, fear, anger, anxiety

Valence classes	Emotion
Unpleasant/Low valence	Fear, anger, disgust, sadness, anxiety
Neutral/Medium valence	Surprise, neutral
Pleasant/High valence	Joy and happiness, amusement

evaluate on the first paradigm only, as it is consistent with the other datasets in the sense of using video stimuli to elicit spontaneous responses. During their experiment data collection, multiple cameras were used for recording. We used the view from "close up from the bottom right" camera due to the frequent incompleteness of some subjects' face in the frontal camera. The self-reported emotion category includes neutral, anxiety, amusement, sadness, joy, disgust, anger, surprise and fear. Removing subjects with incomplete annotation gives us 27 subjects. Each subject has 20 segments, which in total last around 40 minutes.

4.4.2 Mapping emotions into arousal and valence

The annotated emotional categories vary across the datasets. For the purpose of analysis and comparison, we apply the mapping suggested in [101] and [29] to convert the categorical emotions into arousal and valence (see Table 4-2). Hence, two 3-class classification problems are defined for DISFA, MSARV, and MAHNOB, i.e. the classifications of low, medium, and high levels of arousal and valence, respectively.

4.5 Experimental Results

We first demonstrate the effectiveness of our feature representation by benchmarking fast-PADMA against the state-of-the-art performances on UNBC and MAHNOB. We then compare different alignment techniques and specifically evaluate the ensemble mechanism across all 4 datasets. This is followed by a summary of different learning paradigms, which shows the advantage of the proposed user-adaptive model.

4.5.1 External comparison on UNBC and MAHNOB

Since most of the pertinent MIL techniques proposed for facial affect recognition were evaluated on UNBC, we first compare fast-PADMA against the state-of-the-art in the user-independent learning paradigm. The AMIL related features are excluded in this evaluation paradigm, due to unavailability of the target data. The performance is measured by the accuracy at the equal error rate, following the previous protocol in [92][98].

Table 4-3 shows the results. Fast-PADMA, which learns from the pooled statistics features of the aligned data, achieves performance equivalent to the best reported state-of-the-art performance (85.7) on UNBC. This result implies that (1) in spite of the simplicity, well-designed geometric facial features can give a high classification performance for pain detection and this is in line with the comment in [97] that geometric feature is not yet fully explored; and (2) the proposed aggregation technique ϕ_s, that extracts the segment attributes from frames by pooling descriptive statistics features is simple, fast and effective.

Table 4-3. Performance and comparison to state-of-the-art MIL methods on user-independent learning, UNBC dataset (Performance metric: accuracy at equal error rate)

MIL-Boost[110]	MILIS [33]	MILES [15]	MS-MIL[99]	AMIL [46]	RMC-MIL[92]	fast-PADMA* (ours)
76.9	76.9	78.2	83.7	84.4	**85.7**	**85.7**

*Segment feature representation in the experiment contains the pooled statistics features only. AMIL is not used due to the unavailability of the target data.

Compared to the UNBC dataset, few studies have been conducted on MAHNOB, probably due to its challenges – the spontaneous affects it contains are more subtle and complex. We, therefore, compare our result with the latest published results in [47], which followed the protocol in [101] to map the categorical emotions into three classes for arousal and valence.

Table 4-4 shows our comparison results. As indicated by the bolded numbers, fast-PADMA outperforms the k-NN and SVM models [47] in both arousal and valence classifications on MAHNOB. This further validates our techniques of segment feature extraction and data alignment.

Table 4-4. Performance and comparison to state-of-the-art MIL methods on user-independent learning, MAHNOB dataset (Performance metric: accuracy at equal error rate)

Arousal			Valence		
EPFKNN [47]	EPFSVM [47]	fast-PADMA* (ours)	EPFKNN [47]	EPFSVM [47]	fast-PADMA* (ours)
46.4	53.6	**58.4**	44.1	50.6	**53.9**

*Segment feature representation in the experiment contains the pooled statistics features only. AMIL is not used due to the unavailability of target data.

4.5.2 Internal comparison results overview

For a comprehensive understanding of the classifiers built on different data and in different manners, Table 4-5 summarizes the performances of four highlighted models using the proposed method. The comparison was conducted among the user-specific classifiers, generic classifiers, the best performing hybrid model (TXS) and the ensemble classifier of multiple weak generic classifiers (EGC). As expected, all learning paradigms outperformed the baseline, i.e. the majority guess. Given the data skew on MSARV-valence (only one instance was labeled as "medium"), this classification problem has a high baseline and it is relatively easy to correctly classify. By contrast, classifications on MAHNOB are much more challenging, due to its data diversity and subtle expression.

More interestingly, the generic classifiers generally reached significantly higher performances than the user-specific classifiers, except for some close matches on MSARV (arousal: 78.3 vs 76.4 and valence: 96.2 vs 94.2). This may be because the proposed data alignment helps to alleviate the personal bias for the generic classifier.

Table 4-5. Performance comparison between different learning paradigms: specific, generic and hybrid models (Performance metric: correctly classified rate)

DATASET / MODEL	UNBC	DISFA		MSARV		MAHNOB	
	PAIN	A	V	A	V	A	V
BASELINE	62.8	45.3	35.8	39.2	69.5	44	39.3
SPECIFIC	72.1	55.6	53.9	76.4	94.3	46.9	43.6
GENERIC	85.7	74.5	72.4	78.3	96.2	58.4	53.9
TXS	86.5	74.6	72.6	78.3	98.1	58.5	55.0
EGC	**92.5**	83.5	79.0	88.7	99.1	63.6	63.4

TXS: the best performing model learned from available target data with a certain percent of the generic data
EGC: the proposed ensemble classifier of multiple weak generic classifiers

As expected, performances of TXS learned from the target user's available data and partial of the source subjects' data surpassed those of the user-specific and generic classifiers. This corroborates our idea that using both target and source data can facilitate affect modeling.

However, TXS contains the uncertainty of the proper amount of source data. It is highly encouraging to see that the proposed ensemble classifiers EGC successfully outperformed the TXS with a remarkable improvement by 10.2 on average. This verifies our assumption that bootstrapping multiple weak generic classifiers and aggregating them according to the performance can establish a well-performing adaptive model in an efficient manner.

4.5.3 Evaluating the individual data alignment and the adaptive model

To provide an in-depth view of the effectiveness of the proposed techniques, Figure 4-4 presents a general comparison across the four datasets.

Lines with different colors denote the different data alignment techniques for individual training subjects. The vertical axis shows the accuracy in term of correctly classified rate. The horizontal axis presents the models learned on a different set of data. "Specific" indicates the model learned from the target user's data only. "+20%", "+40%", "+60%", "+80%" and "+100%" represent the models trained on the particular percentage of data from the source subjects and all available target data. Selection of the source data was determined by random resampling. "Generic" means the generic classifier built without considering the target data.

For simplicity, we refer to the various modes of alignment as (a) *Nrw*: applies the raw representation without normalization; (b) *Nub*: uses unity-based normalization to transform each feature of individual subject to [0,1]; (c) *Nsm*: shifts the values of each feature with its mean aligned to zero; (d) *Nsn*: shifts each feature with its value at the neutral frame aligned to zero; (e) *Nubn*: aligns the data according to (15), which simultaneously accounts for the neutral frame and the boundaries.

Likewise, we use the following abbreviations for models: (a) *TAS*: learnt from the available target data and all source subjects' data; (b) *TXS*: the best-performing model of those trained on available target data and different percentages (0%, 20%, 40%, 60%, 80%, and 100%) of source data; (c) *EIC*: consists of *N* weak individual classifiers; (d)

EGC: composed of N weak generic classifiers, each of which is trained on $N-1$ source subjects.

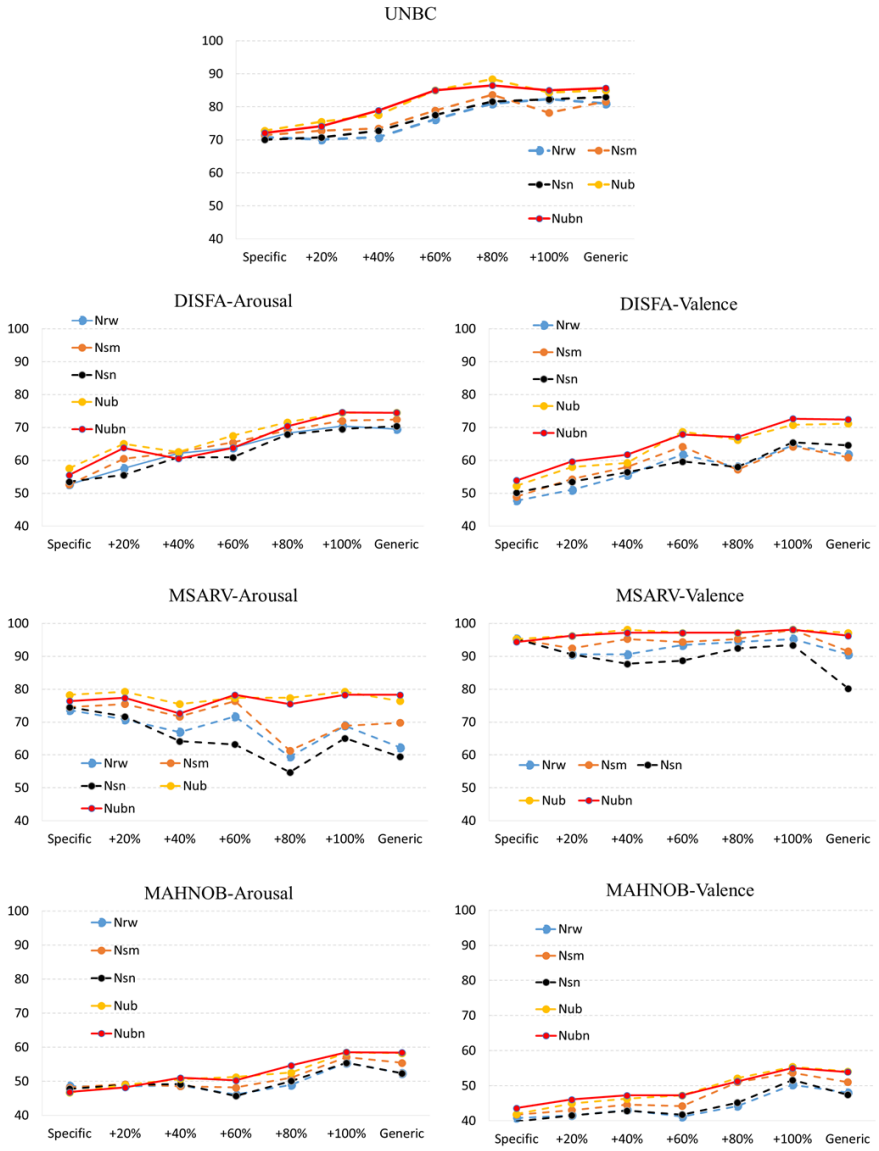


Figure 4-4. Performance in correctly classified rate of pain/non-pain classification on UNBC and three classes' arousal and valence classifications on DISFA, MSARV and MAHNOB datasets. Lines with different colors represent the techniques for data alignment. The vertical axis denote the performance and the horizontal axis the models learnt on different sets of data. "Specific" indicates the model learnt on the target data only. "+20%", "+40%", "+60%", "+80%" and "+100%" denote the models trained on the available target data plus the particular percentage of the source data. "Generic" means the generic classifier built without the use of target data. The overall results show that the proposed data alignment technique (in red) generally performs well among the alignment techniques. In addition, although variations exist among datasets and classification problems, performance usually peaks at hybrid models learnt from target data plus a certain amount of source data.

An overall comparison between alignment techniques shows that Nubn (in red) outperformed others in general. It gave the highest accuracies of arousal on DISFA

(74.6) and MAHNOB (58.5) and valence on DISFA (72.6) and MSARV (98.1). Nubn contributed to a marked improvement compared to Nrw, Nsn, and Nsm, notably for the models trained on additional 80% of source data on DISFA-valence and MSARV-arousal. Figure 4-5 shows some examples of identified neutral frames of different datasets. It is worth pointing out that the unity-based individual normalization (in yellow) also performed well on average, which indicates the effectiveness of individual boundary alignment.

Learning from the incremental data aligned by Nubn normally leads to a steady increase in accuracy. In other words, as we increase the training data from different subjects (observed from the first six data points), the performance of the data alignment with Nubn increases monotonically. This trend suggests that, given a good alignment,

Figure 4-5. Examples of the identified neutral frames from different datasets.

data from different source subjects can be used together in one generic classifier. In contrast, improper alignment of the data from different subjects may hurt the performance of a classifier due to inter-personal differences. Examples can be seen at the performance fluctuation at "+80%" in MSARV-arousal and the decrease from "+20" to "+40%" in MSARV-valence classification.

Upon closer inspection, comparing TAS ("+100%") and the generic classifier ("Generic"), we observe that TAS slightly outperforms generic by 1.8% on average. Judging from these marginal difference, it is highly likely that the personal information is diluted in TAS, as the data ratio between the target and source in most cases tends to be small. This is in line with the results of previous studies that naively combine the target and source data [107], in which it was found that simply adding target data to a training set is not effective to use the limited target data.

4.5.4 Evaluating the ensemble classifier

The aforementioned results validate the effectiveness of the segment feature representation and individual data alignment. In addition, the results also indicate the potentially favorable performance can be achieved by the hybrid model. However, rather than determine the amount of generic data for training, this section demonstrates that the ensemble weak generic classifiers can be an efficient and promising choice.

Figure 4-6 presents performance comparison across learning mechanisms and data alignment techniques. Each bar represents an accuracy of a model, and each group denotes one alignment technique.

Although some exceptions existed on UNBC and mainly on MSARV, TXS performed similarly equally to TAS across the board. This equivalence is in line with our hypothesis of the small training data ratio between the target and source subjects. This ratio for MSARV, however, is relatively high, since MSARV contains a good amount of data per individual and few subjects. Therefore, it is less likely the target information is diluted in the source data, leading to the different result between TAS and TXS. But in general, an approach to adopting the generic commonality while maintaining the target information is needed.

In line with the spirit of inductive transfer learning [13], EIC reweighs the weak individual classifiers based on the accuracy on the available target set. However, EIC, on the whole, failed to compete with either TAS or TXS, shown by the first three bars

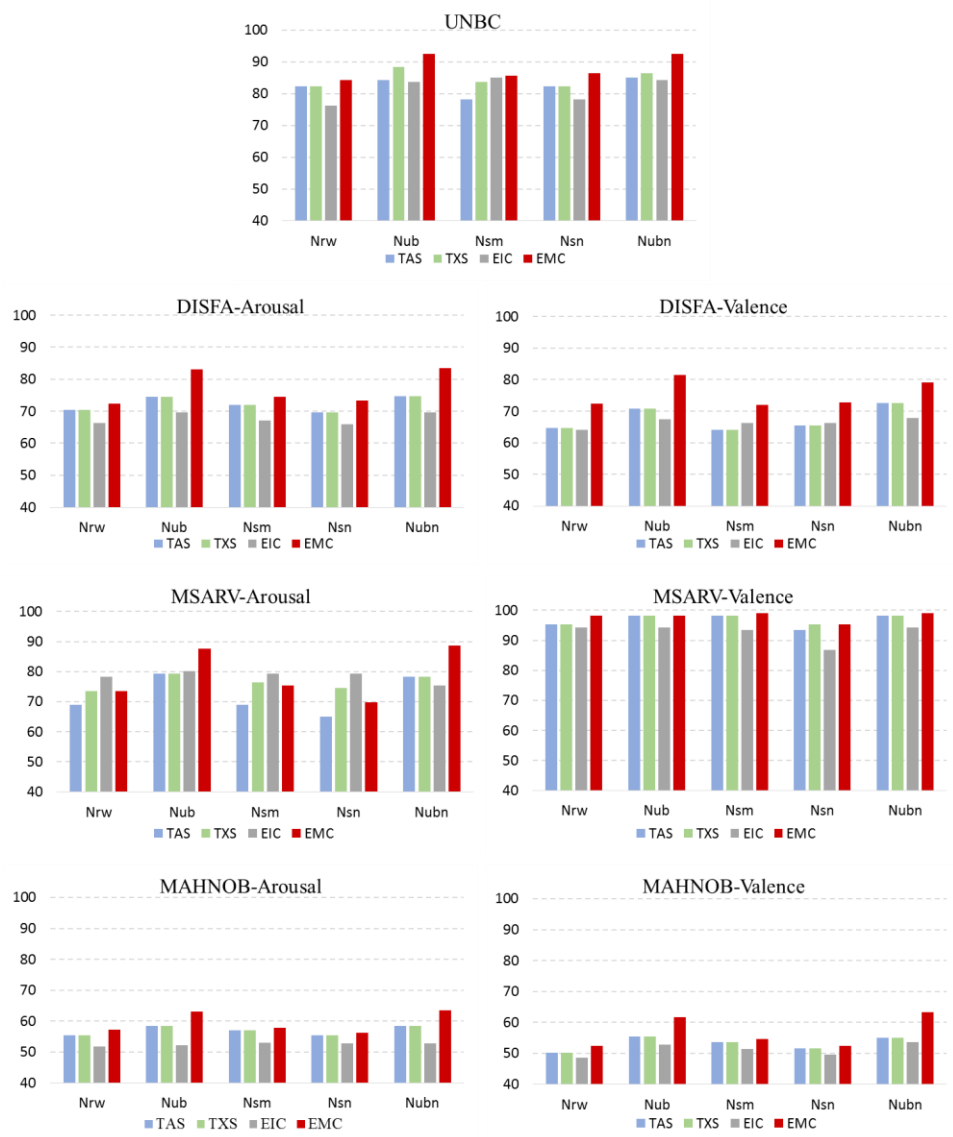


Figure 4-6 Performance comparison in correctly classified rate: performance trained on all available target and source data (TAS), best possible performance trained on all target data and some percentage of source data (TXS), performance of the ensemble classifier of N weak individual classifiers (EIC), performance of the ensemble classifier of N weak generic classifiers trained on multiple individuals (EGC). Across different datasets and data alignment techniques, EGC with Nubn in general outperform other counterparts, which confirms again the validity of the alignment technique and indicates the effectiveness of the proposed ensemble technique.

of each subfigure in Figure 4-6. This shows that the weak individual classifiers can be vulnerable and a low-cost straightforward aggregation may not be satisfactory.

In contrast, it is encouraging that the proposed ensemble classifier EGC substantially outperformed the counterparts. Comparing two ensemble classifiers using the same data alignment (Nubn), EGC surpassed EIC by 8.2 on UNBC, 14.0, 13.2, 10.8 for arousal on DISFA, MSARV, and MAHNOB, and 11.1, 4.7, 9.7 for valence. Inspecting the data reveals that the improvement of EGC over TXS is less obvious or even non-

existent for some alignments, such as using Nrw, Nsm and Nsn on MSARV-arousal. Given the proposed alignment technique (Nubn), however, the improvement is consistent across datasets. Since EGC is composed of multiple generic classifiers, it is not difficult to see that an effective individual data alignment technique is critical to this ensemble classifier. It is worth noting that the proposed Nubn reached a higher or equal performance in six out of seven cases compared to the second best alignment technique Nub, including UNBC, DISFA-arousal, MSARV-arousal, MSARV-valence, MAHNOB-arousal, and MAHNOB-valence.

The proposed EGC with Nubn shows promising results. For a better understanding of the performance, we provide further evaluation focusing on the ensemble learning mechanism.

The common ensemble learning methods include bagging, boosting, random subspace, and stacking. Bagging, also known as bootstrap aggregation, trains multiple models on the random drawn of training subsets and aggregates them with equal weight. Boosting incrementally trains and reweighs the previously misclassified instances. Random subspace method combines classifiers learned from the subspaces of the original feature space. And stacking is to mingle the results from different types of classifiers. While Adaboost [31] is a classic boosting algorithm, random trees is a well-used random subspace method, and random forest [8] is the bagging version of random trees. We therefore compare the following ensemble methods: (1) bagging and (2) Adaboost with SMO, (3) random tree and (4) random forest, and (5) stacking of SMO & random forest, (6) SMO & k-NN [3], and (7) random forest & k-NN. The SMO classifiers in bagging, Adaboost, and stacking were configured with the identical parameters as the previous evaluation.

Figure 4-7 compares the performance of EGC and the other ensemble learning methods on 4 datasets. Surprisingly, except for the relatively poor performance of random tree, different ensemble methods generate approximately the same results. It is encouraging to observe that the performances of our EGC are in general considerably higher than other ensemble methods for different classification issues. On average, EGC outperformed the best counterparts by 5.5%. The only marginal match comes from the classification of MSARV-valence, where the baseline is originally very high.

In sum, the presented experimental results demonstrate the effectiveness of the geometric facial features and segment-level feature representation by comparing to the state-of-the-art performances on the public datasets. Examining the different training

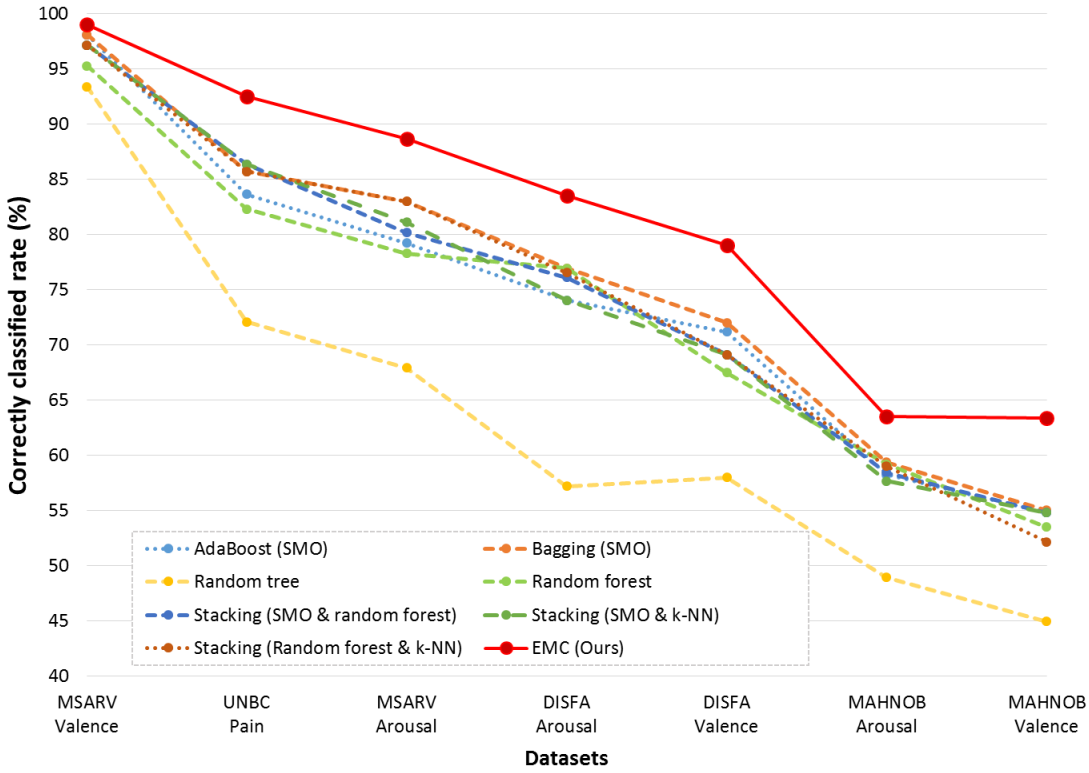


Figure 4-7. Performance comparison of different ensemble learning classifiers. The vertical axis denotes the correctly classified rate. The horizontal axis shows the classification issues on different datasets, which are ranked according to the average performance.

data ratio between target and source data indicates the potential advanced performance of the hybrid model over both the user-specific and generic counterparts. And the comparison among models with different individual data alignment techniques and ensemble methods shows the validity of the proposed alignment technique and the effectiveness of the ensemble model in fusing target and source data.

4.6 Summary

This chapter delineates the relevant techniques to facilitate the rapid modeling of the user-adaptive facial affect model, including the segment-level feature representation, individual data alignment, and the ensemble learning mechanism. Our method learns from both the source and target users. Compared with conventional transfer learning, it relies on no computational distribution estimation to measure the individual similarity, instead, an ensemble of the weak generic classifiers is proposed to learn the commonality from the source knowledge and simultaneously accommodate the identity bias. Experimental results also showed the effectiveness of the segment feature

representation and the validity of the data alignment technique in supporting the model aggregation for ensemble learning.

Our finding conduces to the human-computer interaction by rapidly modeling the facial affect in a practical fashion. Relying on the AMIL technique, our method requires no expertise of annotation. Judging from the evaluations on four public datasets, the proposed approach presents a promising potential in the real-use affect-involved applications.

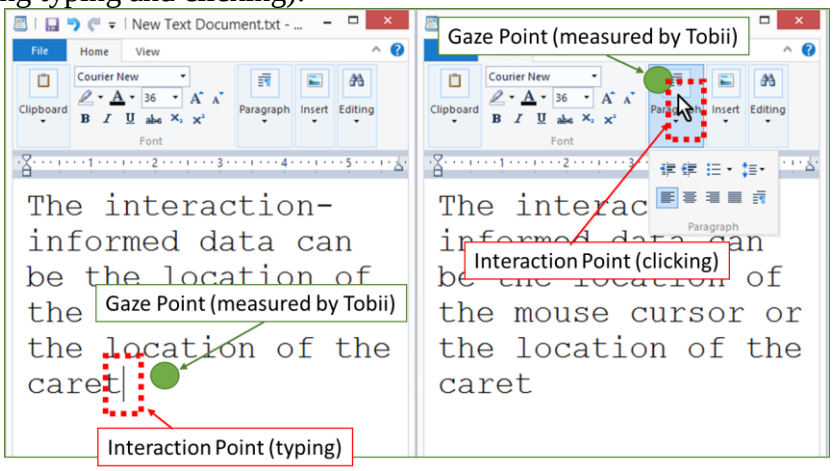
Chapter 5

PACE - Personalized, Auto-Calibrating Eye Tracker

Selected notations and abbreviations used in this chapter

$\boldsymbol{e}$	set of eye features, including $\boldsymbol{e}_r, \boldsymbol{e}_l$ for the right and left eye, respectively
$\boldsymbol{F}$	gaze-point/ gaze-click feature vectors in a 3-second gaze-point/ gaze-click window
$\boldsymbol{f}_j$	the *j-th* column of $\boldsymbol{F}$, i.e. the sequential data of the *j-th* gaze-point/ gaze-click feature in the time window, $\boldsymbol{f}_j \in \Re^{m\times 1}$
$\hat{\boldsymbol{f}}_j$	filtered sequential data of the *j-th* gaze-point feature in the time window
$\boldsymbol{g}_c$	interaction point, i.e. location of cursor or caret
$\boldsymbol{g}_t$	gaze point measure to Tobii tracker
$\boldsymbol{g}_w$	estimated gaze point by the webcam data
M	number of facial landmark, =66
m	number of frame in the gaze-point/ gaze-click window prior to an interaction event, =300
n	number of gaze features, =12 for gaze-point feature vector and =6 for gaze-click feature vector
$\boldsymbol{p}$	a set of parameters for the point distribution model
$\boldsymbol{q}$	non-rigid transformation vector of PDM
$\boldsymbol{R}$	three-axis orientation parameters of PDM
s	parameter of head scaling of PDM
$\boldsymbol{t}$	global head translation parameters of PDM
t_p	time preceding an interaction event
t_s	the last moment of $\boldsymbol{S}_s$
γ	distance threshold to measure whether the gaze error is small, =60 pixels
$\boldsymbol{v}$	gaze-point feature vector, including the head and eye gaze features, $\boldsymbol{v} \in \Re^{n\times 1}$
$\tilde{\boldsymbol{v}}$	estimated gaze-point feature vector that corresponds to the aligned moment of the gaze the interaction cues, $\boldsymbol{g}_c$.
$\boldsymbol{x}$	locations of the facial landmarks in a particular frame
$\boldsymbol{x}_i$	the 3D coordinates of the *i-th* facial landmark
λ	distance threshold in the data-driven validation, =240 pixels in our case
ξ	fixation threshold for gaze-click features
$\boldsymbol{\varepsilon}$	a vector of threshold to measure the stationary state of each gaze-point feature
$\boldsymbol{S}_s$	the stationary period reflected by the gaze features

An illustration of the gaze point (measured by Tobii eye tracker) and the interaction point (during typing and clicking):



Chapter 3 and Chapter 4 present the user-specific and user-adaptive approach to build a well-performing model for the target user. In this chapter, we approach the personalized model from a different angle – the implicit and continuous data acquisition.

Depending on the use context, the explicit human annotation may not be necessary for the computer to learn the human states. We take the building of gaze estimation model as an example in this chapter to study this approach. To achieve this, we exploit the connection between human visual attention and interaction behaviors. The key is to identify the temporal and spatial alignments between interaction cues and gaze locations in an automated fashion.

We give details of an in-depth study into the human gaze behavior when interacting with the computer. To enable the implicit learning for the gaze point estimation in the wild and get rid of the need for specialized equipment, we develop the Personalized, Auto-Calibrating Eye Tracker (PACE) method. It is designed to learn from daily noisy interaction-informed data and facilitate the gaze estimation learning using an off-the-shelf webcam system.

To present our findings and demonstrate the effectiveness of the proposed method, this chapter starts by describing the data we collected for the gaze behavior study and the corresponding behavior patterns we identified. We show that the moment of best alignment varies across different interaction activities, the context of each interaction, and the individuality of the users. Following up on these findings, we present the validation mechanism of PACE and demonstrate that PACE can accurately identify the aligned moment between interaction cues and eye gaze automatically. We then present an evaluation of the gaze model learned on another unconstrained dataset for a thorough understanding of the PACE visual attention estimation performance.

5.1 Collecting Gaze and Interaction-Informed Data

Previous work has assumed that there is a strong correspondence between gaze and interaction, that is, people are looking where they click. In contrast, we hypothesize that the moment of strongest gaze-interaction correlation depends on the nature of the interaction, the context, user habits, and preferences. Some activities require a more explicit demonstration of human intention, which results in a higher gaze-cursor consistency. For example, the positions of the eye gaze and the mouse cursor should be

Table 5-1. Examples of gaze patterns from common interaction behaviors (ECS, ECLS, ECL, ED, ET).

Interaction event	Human intention	Potential gaze pattern
Mouse click	Link or button selection	Fixation on the mouse cursor
Mouse double-click	Word selection in document editing	Fixation on the mouse cursor
Mouse button up after drag	Paragraph selection in document editing	Fixation on the mouse cursor
Keyboard letter key down	Word typing	Fixation/smooth pursuit on the typing caret

better aligned during a mouse click event, compared to when the mouse cursor is being used as a reading aid [43]. The context also affects the correlation. It is easy to see that clicking to select a single character in a paragraph of text would require a more purposeful and precise gaze than clicking on photo thumbnails to scroll through photo albums. Some tasks may actually require or encourage the user to look at a part of the screen away from the cursor, e.g. pausing the playback of a video at a precise moment during video editing.

Our investigation, therefore, focuses on four commonly encountered interaction activities, as shown in Table 5-1. We use the Tobii EyeX tracker to provide us with the "ground truth" of the user's gaze point. The Tobii tracker uses active infra-red technology with 60 Hz tracking frequency and an accuracy of within 1º visual angle (corresponding to 30-50 pixels on our 22" monitor at 1680×1050 resolution and a reading distance ≈500~800mm), and can be considered to be state-of-the-art.

We recruited 31 subjects (16 female, ages 20-30 yrs, M=25.1, SD=2.5) for this study. The subjects were university students and staff. 24 of them are capable of touch-typing, at least for the letter keys.

Subjects were asked to work naturally, which meant the free movement of head and body. They were allowed to change the chair position and height, but to keep their head-to-monitor distance within the valid range (≈500~800mm) of the monitor. Three experiments were designed to generate the necessary interaction behaviors:

Correlation between visual attention and click targets: The first experiment requires the subject to click on targets of different shapes and sizes. On the basis of our

preliminary observations of mouse usage, our study focuses on three main types of mouse click targets: (1) long slim targets, (2) small targets and (3) large targets.

To obtain data on (1) and (2), a list of academic papers with long titles (≥ 3/4 of the screen width) was prepared in advance. The subjects were asked to search for each paper on Google Scholar, and to click on the hyperlinks for the title and authors for each paper. Since Google abbreviates authors' first names, author hyperlinks are usually short and button-like (small targets), and the hyperlinks with the paper titles give long slim targets. The large targets were obtained by asking the subjects to search for and click on photos in Flickr that they found interesting. The width of the photos occupied around one-third of the screen.

Correlation between visual attention and mouse drag actions: The second experiment considers a different kind of mouse activity – dragging. Subjects were asked to select sentences from a given PDF document by dragging the mouse. To ensure that they would actually pay attention to what they were doing, they were required to select complete sentences or phrases (ending with a period or comma).

Correlation between visual attention and keyboard usage: For the third experiment, subjects were asked to type a short paragraph into a text file. They could type anything they wanted, as long as it was a syntactically correct paragraph that made sense semantically. We collected only letter keys ("a"-"z", "A"-"Z", spaces), because most people, even those who can touch-type, are actually only able to touch type the letter keys, not the number or function keys.

Each of the experiment subjects was required to generate at least 50 instances of each interaction activity. The occurrence of a clicking event was defined as the press of the left mouse button, dragging events as the release of the left mouse button, and keypresses as the depression of a letter key. Each interaction event triggered a screenshot that was saved for data validation and event classification. In total, we collected 1915 clicking events on long slim targets, 2344 on small targets, 1955 on large targets, 2029 dragging events and 4863 typing events.

5.2 Evaluation of the Correspondence Assumption

To investigate the correlation between visual attention and interaction event, we study the data from the Tobii tracker in every *gaze-point window*, i.e. 3 seconds *preceding* each interaction event. The focus on pre-interaction behavior is informed by previous

work [43], which reports that in general, the position of the cursor lags behind the gaze point, not the other way around.

For simplicity, we will use the following abbreviations when referring to events of: clicking on small targets (ECS), long slim targets (ECLS), large targets (ECL), dragging (ED) and typing (ET).

The eye tracker returns the position of the user's gaze on the screen as a temporal sequence of (*x*, *y*) screen coordinates. To allow for inherent error from the eye tracker, we choose a small distance threshold γ (=60 pixels), which matches the equipment error, i.e. the average estimation error of Tobii eye tracker used for ground truth collection in our experiment setting. The position of the user's gaze and the location of the interaction event are considered to be *aligned* when their *displacement*, or the distance between them, is less than this threshold, *i.e.* $\|\boldsymbol{g}_t(t_p) - \boldsymbol{g}_c\| < \gamma$, where $\|\boldsymbol{g}_t(t_p) - \boldsymbol{g}_c\|$ indicates the Euclidean distance between the tracker-measured gaze point $\boldsymbol{g}_t$ and the interaction point $\boldsymbol{g}_c$, and t_p represents the time preceding an interaction event.

Figure 5-1 shows the probability of *gaze-interaction* (i.e. gaze-cursor or gaze-caret) *alignment* as we approach the moment of interaction, *i.e.* $Pr(\|\boldsymbol{g}_t(t_p) - \boldsymbol{g}_c\| < \gamma)$. The *x*-axis shows the time in seconds *before* the interaction event. As expected, the

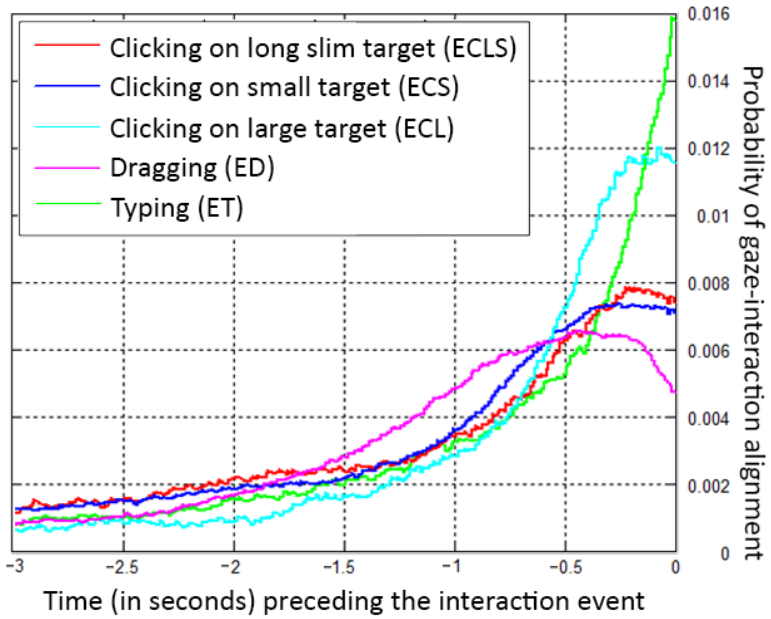


Figure 5-1. Probability of gaze-interaction alignment – i.e. the likelihood that visual attention is spatially located at the interaction event, as a function of time *preceding* the event.

probability of the gaze-interaction consistency generally increases as we get closer to the interaction event. However, the moment of highest probability for gaze-interaction alignment does not necessarily occur at the moment of the interaction. For example, the likelihood distribution of gaze-interaction alignment peaks at t_p=-0.01s for ET, -0.07s for ECL, -0.20s for ECLS, -0.25s for ECS, and -0.43s for ED. In particular, for mouse drag events (ED), the probability of gaze-interaction alignment *falls off significantly* in the moments just before the interaction event happens. It would seem that typing events (ET) are the only ones in which the assumption that "the user is looking where he/she is interacting" is consistently upheld.

Figure 5-2 presents the *gaze-interaction displacement i.e.* $\|\boldsymbol{g}_t(t_p) - \boldsymbol{g}_c\|$ or the distance between the location of the user's gaze and the eventual location of the interaction event. It can be seen that the displacement (mainly: the grey and blue regions) generally decreases as we get closer to the time of the interaction event, but the distributions are quite dissimilar. Unsurprisingly, the displacement is largest for ECL (clicking on large targets). However, the interaction activity with the second largest displacement is ET (typing), which also has a large range of displacement values (wide blue and grey regions). This implies that even though Figure 5-1 suggests that people are *generally* looking at the caret when they type, there is still much variation across different events, and hence, using the raw typing-informed data for gaze model training would introduce much noise and error into the system.

The distribution of the outliers (the red regions) is also of interest. Figure 5-2 shows that there is much variation in the user's gaze. Furthermore, the *range* of these locations is very great, ranging to 1000 pixels away from the interaction position. This further corroborates our hypothesis that the correspondence assumption is not valid in real-use situations. This means that a naïve use of raw interactional-informed data for gaze model learning is not likely to produce optimal results.

Inspecting the non-outlier data reveals some interesting findings. For ED (dragging), a U-shape distribution starts to form around 1 second before the interaction event. This indicates that in most cases, subjects start looking elsewhere before the drag event is complete. A similar phenomenon happens during ECLS. This suggests that if we could identify the point of least displacement, this would potentially be a better indicator of eye gaze than simply collecting the data at the moment of the event.

Another interesting finding comes from the *densities* of the displacement distributions across the different behaviors. Although the *range* of the displacements, *i.e.* the upper boundaries of the gray region, is fairly large, 75% of the data stays within half of the range, as evidenced by the fact that the upper boundary of the blue regions lies close to the middle of the gray regions. Similarly, the green median line lies below the middle of the blue region for almost all behaviors, which indicates that the data is very compactly distributed. This is especially true for typing events (ET). This implies

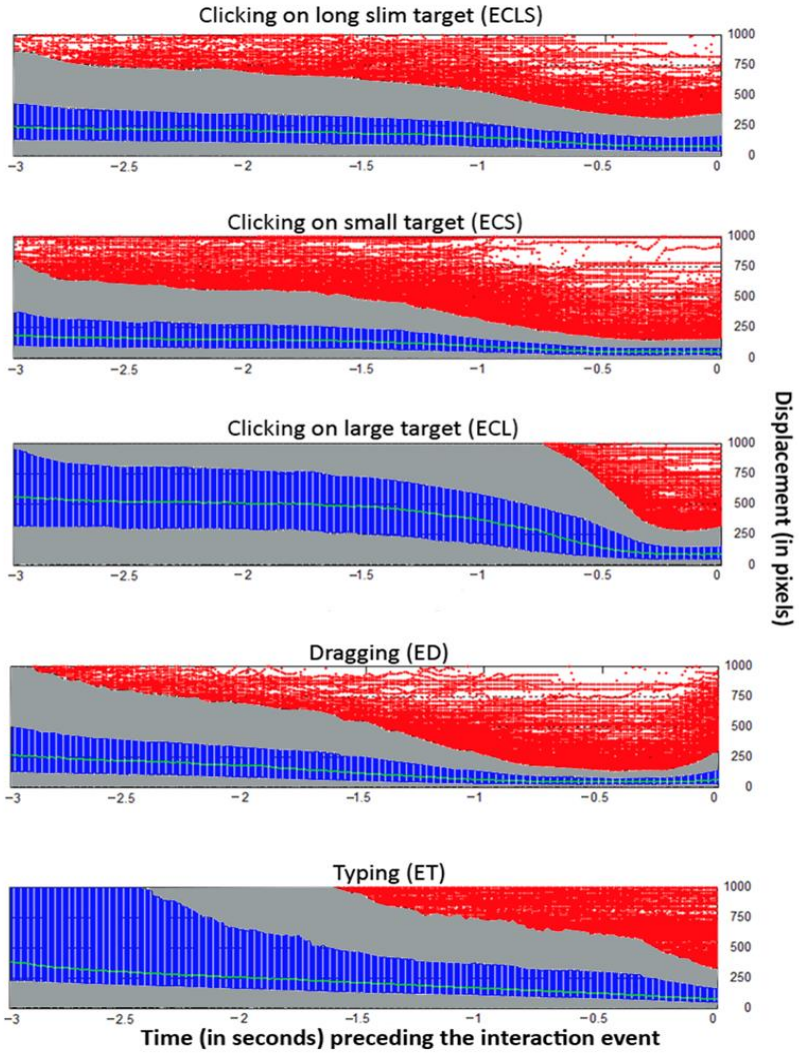


Figure 5-2. Displacement between gaze and interaction cues (cursor/caret) as a function of pre-interaction event time for different interaction activities. The *x*-axis indicates the time before the event, the *y*-axis shows the Euclidean distance between the location of event (i.e. where the mouse is actually clicked or where the character appears on the screen) and gaze coordinates collected by the eye tracker. The green line shows the median distance. The blue region shows values that fall within $[p_{25}, p_{75}]$, *i.e.* the 25th and 75th percentiles. The grey area indicates the range of the data points that are not considered as outliers. The red points are the individual outliers, defined as values located beyond $[2.5p_{25} - 1.5p_{75},\ 2.5p_{75} - 1.5p_{25}]$.

that the majority of the data exhibits strong gaze/interaction consistency, even in real-use situations.

In addition to the data analysis, observations of the subjects' behavior during the experiment and the post-experiment interviews also reveal some interesting insights and provide possible explanations for the data distribution.

Clicking: We observed two distinct ways in which people usually click on long slim links. Some people start moving the cursor towards a link only after reading its context and deciding to click on it. In this case, the user's gaze tends to stay close to the last word they read and that is usually also where they click. However, sometimes the user perceives that there is a high probability that a certain link would be relevant, even before reading it. In these cases, they often move the cursor to hover over the link before actually reading the words. Once they finish reading, they click the mouse without moving it again. In these cases, the displacement between gaze and click could be quite unpredictable, depending on where the cursor hovers and where the link ends.

Dragging: Dragging generally results in high consistency between gaze and interaction-informed data. Since the nature of the action requires precision, people are more likely to spend more time and care to ensure that the context selection is correct. However, by the time a drag has been completed, users may already be looking for the next target, such as the "highlight" button. This may be the main reason that the probability of gaze-cursor alignment drops as we get closer to the moment of the event as shown in Figure 5-1 and that the corresponding large displacement causes the U-shape distribution as shown in Figure 5-2. The context is also important. When the selected sentence ends a paragraph, the PDF viewing application automatically snaps the end of the drag to the end of the paragraph. Therefore, subjects are often more careless when selecting such sentences, thus creating a large distance displacement.

Typing: For users who can touch type, their gaze normally follows the caret, *i.e.* the location of the character being typed. However, we observed that when they are thinking hard, many (11 out of 24 in our case) touch-typers look elsewhere on the screen while continuing to type. For users with limited touch-typing skills, the gaze switches between the monitor and keyboard. Both of these behaviors create large displacement and explain the presence of a large number of outliers for this interaction activity.

The above investigation corroborates our hypothesis that the moment of strongest gaze-interaction correlation contains a high degree of uncertainty due to the impact of interaction attribute, context, user habits, and preferences. An automated approach to

identifying the gaze-interaction alignment is imperative to make use of such daily interaction-informed data for gaze learning.

5.3 Estimating the Location of Visual Attention

The findings from our gaze behavior study show that there are moments in which gaze-interaction alignment are more likely. We hypothesize that it is possible to use *knowledge of common gaze patterns* and *analysis of visual signals* to identify the point at which user gaze and interaction event are most likely to be aligned. In other words, we postulate that there are periods when user gaze-interaction event alignment is likely, and it is possible to detect these periods in an automated fashion.

5.3.1 Designing the framework for implicit gaze learning

Figure 5-3 gives an overview of our PACE approach. A standard webcam captures video of the user's head and shoulders while mouse movements and keystrokes are tracked by the system. Two tracking models extract the gaze-point feature vector $\boldsymbol{v}$ from face and eye landmarks identified in the frames of the video stream.

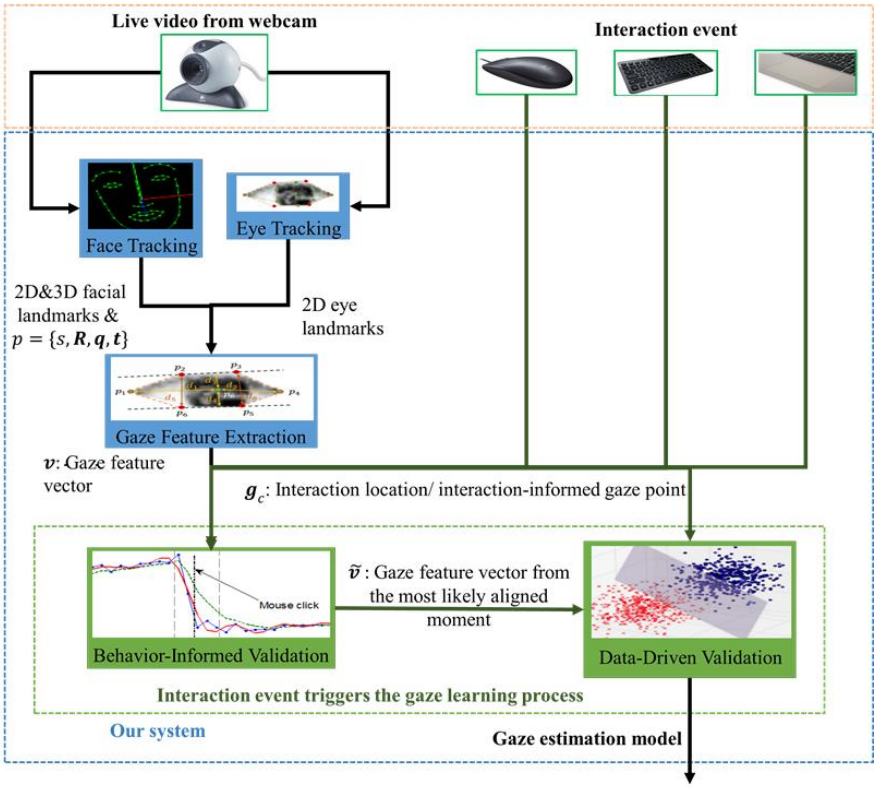


Figure 5-3. Overview of the PACE methodology: combining interaction data and webcam video for eye gaze modeling.

Upon the trigger of an interaction event, gaze-point feature vectors from the 3-second gaze-point window preceding the interaction are sent to a *behavior-informed validation* engine and a *data-driven validation* engine. The behavior-informed validation engine selects one vector $\tilde{\boldsymbol{v}}$ that corresponds to the moment when the user's gaze is most likely to be aligned with the interaction event. The data-driven validation engine then further checks the validity of $\tilde{\boldsymbol{v}}$, based on the previous training samples. If $\tilde{\boldsymbol{v}}$ passes both of these validation steps, $[\tilde{\boldsymbol{v}}, \boldsymbol{g}_c]$ will be used as training data to update the gaze estimation model. Otherwise the data is retained for re-evaluation after the next update.

5.3.2 Extracting gaze-point features from video

Head pose estimation

To obtain accurate locations of the facial landmarks and head pose information, we apply Saragih's Constrained Local Model (CLM) [96]. CLMs generate parametric models for face alignment based on the point distribution model (PDM) [22] and the localized patch model, which reflect the interdependency and the local appearance of the landmarks, respectively.

We use a 3D CLM model in our experiments. The 3D vertices of the facial landmarks can be represented by $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)^T$, where $\boldsymbol{x}_i = (x_i, y_i, z_i)$ denotes the coordinate of the *i-th* landmark in the camera coordinate and M(=66) is the number of facial landmarks. Procrustes alignment [21] is used to normalize the rigid transformation and Principal Component Analysis (PCA) is applied to approximate the non-rigid deformation of the given samples. Therefore, the face images and corresponding annotated landmarks can be represented by $\boldsymbol{x}_i(p) = s\boldsymbol{R}(\bar{\boldsymbol{x}}_i + \boldsymbol{\Phi}_i\boldsymbol{q}) + \boldsymbol{t}, i \in [1, M]$, where $\boldsymbol{x}_i(p)$ and $\bar{\boldsymbol{x}}_i$ represent the 3D and mean coordinates of the *i-th* landmark, respectively; $\boldsymbol{\Phi}_i$ the sub-matrix of the basis of variation of the *i-th* landmark; and $\boldsymbol{p} = \{s, \boldsymbol{R}, \boldsymbol{q}, \boldsymbol{t}\}$ the PDM parameters: a head scaling s, a three-axis orientation $\boldsymbol{R}$, a global head translation $\boldsymbol{t}$ and a non-rigid transformation vector $\boldsymbol{q}$.

Generally, the objective of CLM fitting is, given an image $\boldsymbol{\mathcal{I}}$, to determine the PDM parameter $\boldsymbol{p}$ so as to minimize the misalignment of the landmark with the cost function $f(\boldsymbol{p}) = \mathcal{R}(\boldsymbol{p}) + \sum_{i=1}^{M} \mathcal{L}_i(\boldsymbol{x}_i; \mathcal{I})$. This can also be viewed as maximizing the likelihood of both the global geometric shape deformation $\mathcal{R}(\boldsymbol{p})$ and the local patch textural information of the landmarks, $\sum_{i=1}^{M} \mathcal{L}_i(\boldsymbol{x}_i; \boldsymbol{\mathcal{I}})$. However, due to the nature of the training

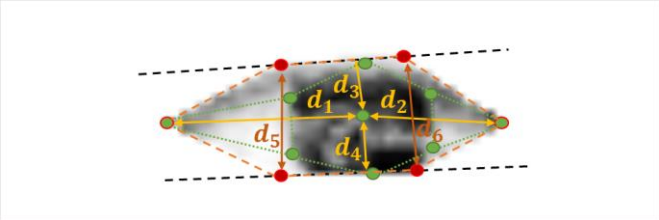


Figure 5-4. The eye landmarks in the eye tracking model.

image data and the influence of local maxima in optimization, CLM tracking tends to fail under poor lighting condition and certain kinds of expressions, e.g. mouth corner depression. To improve localization accuracy of the landmarks and to facilitate head pose estimation, we use the Supervised Descent Method (SDM) to optimize 48 facial landmarks [118]. Unlike CLM, SDM learns the sequence of descent directions for individual landmarks from training data, by minimizing the following function over the descent direction $\Delta \boldsymbol{x}$, $f(\boldsymbol{x}_c + \Delta \boldsymbol{x}) = \|\mathcal{A}(\boldsymbol{x}_c + \Delta \boldsymbol{x}) - \mathcal{A}_*\|_2^2$, where $\boldsymbol{x}_c$ denotes the initial configuration of the landmarks, $\mathcal{A}$ represents the appearance feature extraction and $\mathcal{A}_*$the appearance feature near the manually annotated landmarks. During tracking, SDM optimizes the landmark localization with the learnt descent directions.

Our approach uses SDM for fast and robust face tracking. For each frame, we use the SDM fitting results as the initial state for CLM fitting. This improves the localization accuracy and reduces the time needed for convergence. CLM optimization then provides us with the 2D and 3D facial landmarks and the parameters $\boldsymbol{p}$. By removing the rigid geometric variation (s, $\boldsymbol{R}$ and $\boldsymbol{t}$), the similarity-normalized shape, i.e. the aligned 3D landmarks, can be obtained.

Pupil center estimation

An accurate estimation of gaze direction requires knowing the location of the center of the pupil. However, in unconstrained situations, the appearance information of the eye region usually fails to provide a clear cue for the pupil center. Additionally, we observe that quite often in real-use scenarios, low video resolution, and reflections on glasses and cornea may make the region of the pupil and its periphery unobservable.

To address issues caused by the unconstrained environment and noisy images with low resolution, we use eye geometric features to estimate the pupil center. The general idea is to track landmarks with good salient features on the iris contour and eyelid corners, and deduce the pupil center locations based on their geometric interdependency. We, therefore, apply eye CLMs to track the eye landmarks, including

the pupil center (Figure 5-4). To validate the eye tracking state, we apply a robust but rough tracking model using the integral image [83] to approximate the pupil center from the eye image, with the assumption that the darkest rectangular area is the iris. If the CLM estimation is too distant from the integral image approximation, we assume that the CLM tracking has failed and we redetect the eye for tracking.

Based on the landmarks identified by the face and the eye trackers, we extract n (=12) features and construct a *gaze-point feature vector* $\boldsymbol{v} = [s, \boldsymbol{R}, \boldsymbol{t}, \boldsymbol{e}_r, \boldsymbol{e}_l]^T$, $\boldsymbol{v} \in \Re^{n\times 1}$, where $s, \boldsymbol{R}, \boldsymbol{t}$ are the head pose features obtained from CLM, while $\boldsymbol{e}_r, \boldsymbol{e}_l$ denote the eye features from both eyes, each of which is defined according the eye landmark distances shown in Figure 5-4 as

$$\boldsymbol{e} = \langle \frac{d_1}{d_1 + d_2}, \frac{d_3}{d_3 + d_4}, d_5 + d_6 \rangle \tag{5.1}$$

5.3.3 Using human behavior to inform data validation

The well-observable human gaze patterns by webcam system can be categorized into four behaviors: *fixation*, *smooth pursuit*, *saccade* and *blink* [39]. Fixation indicates a stationary gaze. Smooth pursuit denotes relatively slow gaze movements, while saccades are eye movements that rapidly direct towards a stationary target. Figure 5-5 shows the change in an example feature signal (eye yaw) surrounding a mouse click. The gaze pattern contains 2 short fixations, 2 saccades, and 1 smooth pursuit. The black dashed line indicates the moment of the mouse click. The figure shows that the user's gaze was originally focused on one point (1st fixation), then rapidly moved towards a short link (1st saccade), which took approximately 1 second to read (smooth pursuit). She then clicked on the link and her attention shifted (2nd saccade) elsewhere (2nd fixation). This behavior is consistent across multiple instances, with the mouse click usually occurring at the end of the smooth pursuit, the beginning of the 2nd saccade or

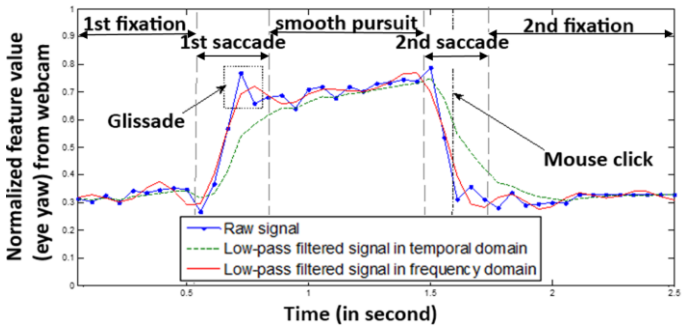


Figure 5-5. Raw and filtered webcam signals from a sample mouse click event (user is reading from left to right).

even the beginning of the 2nd fixation. This clearly illustrates how the user's fixation is not always located at the point of the interaction event.

The behavior-informed validation in PACE identifies moments at which the user's gaze aligns with the cursor/caret when interaction activity is triggered. Based on knowledge of human gaze patterns and our observations, this is most likely during periods of fixation or smooth pursuit. For simplicity, we refer to these periods as $\boldsymbol{S}_s$ and the state of the gaze-point feature during these periods as being "stationary with small trend", or "stationary". The problem then is to determine $\boldsymbol{S}_s$ automatically from the webcam data.

Signal smoothing and filtering

Based on our previous findings, we focus on the user signals collected in the 3 seconds prior to the interaction event. Linear interpolation is used to resample the signal to 100Hz, giving us 300 samples per feature per interaction event. When an event is captured, we use the feature vectors in the previous m(=300) frames prior to the event to construct the gaze-point feature matrix $\boldsymbol{F} \in \Re^{m \times n}$. Since the raw webcam signals contain much high-frequency visual noise, low-pass filtering in the frequency domain is applied to remove the high frequency temporal jitter from the signal while maintaining the main component without overmuch distortion. The filtered signal is calculated as

$$\hat{\boldsymbol{f}}_j = \mathcal{F}^{-1}\left[\mathcal{F}\left(\boldsymbol{f}_j\right) * \mathcal{H}(\omega)\right] \tag{5.2}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ indicate the one-dimensional forward and inverse Discrete Fourier transform, respectively, and the filter

$$\mathcal{H}(\omega) = \begin{cases} 1, & \omega \leq \omega_h \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

where ω_h (=2.5Hz in our work) is the cutoff frequency.

Figure 5-5 demonstrates the effects of filtering in the frequency and temporal domains. While both filtering methods remove much of the high-frequency noise, filtering in the temporal domain results in the loss of some critical information, such as the dynamic overshoot glissade that occurs before the 1st rapid saccade. We, therefore, filter in the frequency domain to remove the high-frequency temporal jitter while maintaining the shape of the main component with minimal distortion.

Extracting a stationary feature vector

Table 5-2. Adaptively extracting a stationary feature vector.

Input: Matrix $\boldsymbol{F}$ of gaze-point feature vectors from the 3-second gaze-point window before the interaction event.	
Output: The stationary period $\boldsymbol{S}_s$ and the feature vector $\tilde{\boldsymbol{v}}$ at the most likely moment of gaze-interaction alignment.	
1	Calculate the low-pass frequency-domain filtered signal $\hat{\boldsymbol{f}}_j$ for each gaze-point feature signal $\boldsymbol{f}_j$
2	Iteratively search for the stationary period using an incremental threshold $\boldsymbol{\varepsilon} \in [\boldsymbol{r}/1000, \boldsymbol{r}/100]$, where $\boldsymbol{r} \in \Re^{n\times 1}$ is the range of each feature over the gaze-point window.
3a	Initialize the overall stationary period with the frame index $\boldsymbol{S} = [1, \dots, m]$.
3b	Calculate the overall stationary period $\boldsymbol{S}$ according to Equation (7.4); set $\mathcal{S}_i$ to 0 if the condition is not satisfied.
3c	Backward search $\boldsymbol{S}$ for the first series of consecutive frame indices $\boldsymbol{S}_s$ whose corresponding duration is longer than the minimum fixation duration (80ms) [39].
3d	**If** t_s, the last moment of $\boldsymbol{S}_s$, occurs within 0.5 seconds before the event, then break. **else** increment $\boldsymbol{\varepsilon}$ by $\boldsymbol{r}/1000$.
	end
4	Perform line fitting for each feature signal $\hat{\boldsymbol{f}}_j$ in $\boldsymbol{S}_s$ to approximate the final gaze-point feature vector $\tilde{\boldsymbol{v}}$, according Equation (5.5).

To identify the stationary period $\boldsymbol{S}_s$ and the corresponding estimated feature vector $\tilde{\boldsymbol{v}}$, our approach adaptively searches for candidate periods occurring close to the point of the event that do not exhibit high signal variance.

To identify these candidate periods, PACE uses a novel adaptive method that searches for a *relatively stationary* period close to the event. The algorithm is similar to our previous proposed method [45], which uses a threshold to determine if the signal change is small enough to be considered "stationary", but that approach requires a truly fixed gaze and will fail if the user's gaze is not truly still, which would be problematic for real-world contexts where the eye is rarely truly stationary.

Table 5-2 presents our new algorithm. Basically, we analyze the changes between consecutive frames in the 3-second gaze-point window prior to the interaction event.

An adaptive threshold $\boldsymbol{\varepsilon}$, which is based on the range of each feature, is used to identify candidate frames whose feature vectors satisfy the following condition:

$$\prod_{j=1}^{n} H(\varepsilon_j{}^2 - \dot{\hat{f}}_{ij}^2) = 1 \tag{5.4}$$

where $H(x) = (1 + \mathrm{sgn}(x))/2$ is the Heaviside step function and $\dot{\hat{f}}_{ij}$ is the *i-th* frame value of the derivative of $\hat{\boldsymbol{f}}_j$ with respect to the sample time. These frames are considered to be potentially within the stationary period.

After all frames in the window have been tested, a backward search is performed to locate $\boldsymbol{S}_s$. Linear regression is used to approximate the gaze-point feature vector $\tilde{\boldsymbol{v}}$ corresponding to the last moment of $\boldsymbol{S}_s$, which under our assumptions is also the moment when the user's gaze is most likely to be aligned with the interaction event. If no fixation or smooth pursuit is detected at all during the gaze-point window, that interaction event is considered to be not suitable for training and is discarded.

Since features in the stationary state are relatively stable, we apply linear regression to approximate the final gaze-point feature vector $\tilde{\boldsymbol{v}}$. M-estimator is used to iteratively fit the line using the weighted least-squares algorithm to minimize

$$\sum\nolimits_{i \in \boldsymbol{S}_s} \rho(u_{ij}) \tag{5.5}$$

where $\rho(u) = 2(\sqrt{1 + u^2/2} - 1)$ is the distance function and u_{ij} denotes the difference between $\hat{f}_{ij}$ and its fitted value $\tilde{f}_{ij}$. Setting k as the last frame of $\boldsymbol{S}_s$, the resulting value for the *j-th* gaze-point feature should be $\tilde{f}_{kj}$. This generates a final input feature vector for the classifier, $\tilde{\boldsymbol{v}} = [\tilde{f}_{k1}, \ldots, \tilde{f}_{kn}]^T$. Two examples gaze-point feature

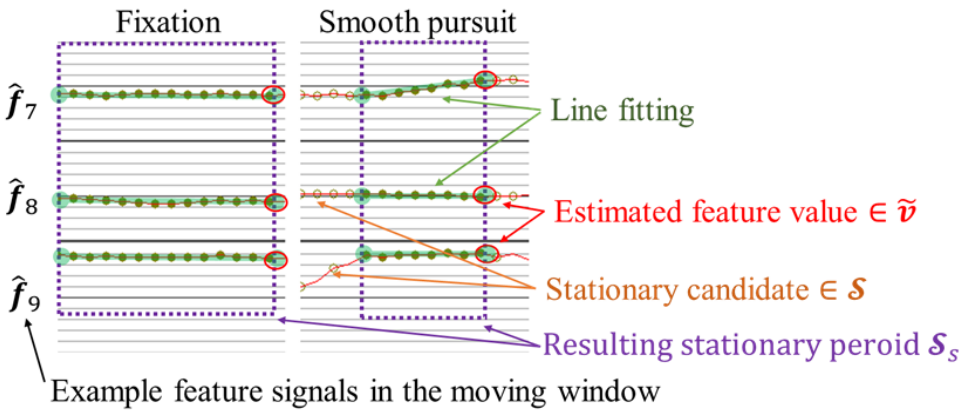


Figure 5-6. Examples of gaze-point feature vector estimation during fixation and smooth pursuit.

estimation are illustrated in Figure 5-6 during a fixation and smooth pursuit, respectively.

Data-driven validation

The behavior-informed validation looks for the stationary period corresponding to the received interaction event. However, there is a possibility that even though the user's gaze is fixated on something, the gaze point may not be anywhere near the location of the interaction event – for example, when the user is watching a movie with the mouse pointer poised over the "pause" button, or typing a chat message while reading the previous incoming responses. To accommodate these types of interactions, PACE uses data-driven validation as an additional layer of validation to determine the goodness of the feature vector $\tilde{\boldsymbol{v}}$ and the corresponding assumed interaction-informed gaze point $\boldsymbol{g}_c$ based on previously validated data.

We use random forest [8] to build the gaze regression models for both x- and y-coordinates. The gaze-point feature vectors $\tilde{\boldsymbol{v}}$ are used as features and the corresponding interaction points $\boldsymbol{g}_c$ as the "truth". Each model generates 100 trees and each tree considers 4 random features. The initial model is trained on the first 100 interaction instances in which the moment of the interaction occurs within a fixation period.

When the most-recently collected feature vector $\tilde{\boldsymbol{v}}$ is passed through the gaze model, it outputs a webcam estimated gaze point $\boldsymbol{g}_w$. If $\|\boldsymbol{g}_w - \boldsymbol{g}_c\| \leq \lambda$, ($\lambda$ = 1/12 of the screen diagonal length in our work and $\boldsymbol{g}_c$ is the interaction-informed point), the instance $[\tilde{\boldsymbol{v}}, \boldsymbol{g}_c]$ is considered validated and can be used as training data. For efficiency, the gaze model is updated in a batch mode; upon the collection of 150 valid instances of new training data, the random forest regression model is retrained on all the validated data. To make full use of all potential data, instances that fail the current validation are also retained and re-evaluated after each update of the gaze model.

In summary, PACE uses a dual-level validation. First, by looking for the stationary gaze-point features corresponding to an interaction event, the behavior-informed validation uses knowledge of gaze movement patterns and user interaction to identify the moment when user gaze and interaction point are most closely aligned. The data-driven validation further applies prior knowledge of the particular user to account for user individualities and contextual differences.

5.3.4 Correctness of assumptions on user gaze patterns

The PACE approach makes some assumptions about the characteristics of user gaze patterns to identify good data points that can be used to train a gaze model. We are interested in whether these data points really correspond to moments when user gaze aligns with interaction event.

This evaluation is conducted on the dataset described in Section 5.1. We use the eye gaze coordinates collected by the Tobii EyeX tracker as the gold standard. For each interaction event, we compare its location $\boldsymbol{c}$ with the Tobii-measured gaze coordinates $\boldsymbol{g}_t$ at the moment identified by the behavior-informed validation mechanism as the point of closest alignment. The error of our PACE method, $error_{PACE}$, is calculated as the displacement (Euclidean distance) between $\boldsymbol{g}_t$ and $\boldsymbol{g}_c$. We also calculate $error_{naive}$, which is the error that would result if we take the naïve assumption and simply choose the moment of the interaction event as the moment of best alignment, as in previous work [103]. As another point of reference, we compute $error_{min}$, which is the minimum possible error that we could achieve if we somehow knew the exact moment of best gaze-interaction alignment in the 3-second gaze-point window; and $error_{minf}$, which is the minimum error achieved after discarding the obvious errors, defined as points where the displacement is larger than 1/12 of the screen diagonal length (=240 pixels). These usually correspond to instances in which the user is clearly not looking at the interaction point – for example, when he/she is looking for a key on the keyboard.

Figure 5-7 compares the performance achieved by the three different approaches on the data described in the previous link clicking and typing experiment. The color bars indicate the displacement values, and the circles denote the percentage of data that is retained after outliers are removed.

The results are encouraging. As expected, $error_{min}$ (blue) can be very small, with an average of 41.1 pixels over all mouse interactions. For keyboard events, $error_{min}$ is much larger at 401.8 pixels. This is caused by the instances in which the user was not looking at what was being typed at all – i.e. the gaze was either wandering about the screen, or he/she was looking at the keyboard. When outliers are discarded, the keyboard event error decreases to 86.7 pixels. Discarding outliers for all events brings the error down to 27.8 pixels. However, this is the best achievable result, which is extremely difficult to achieve in practice.

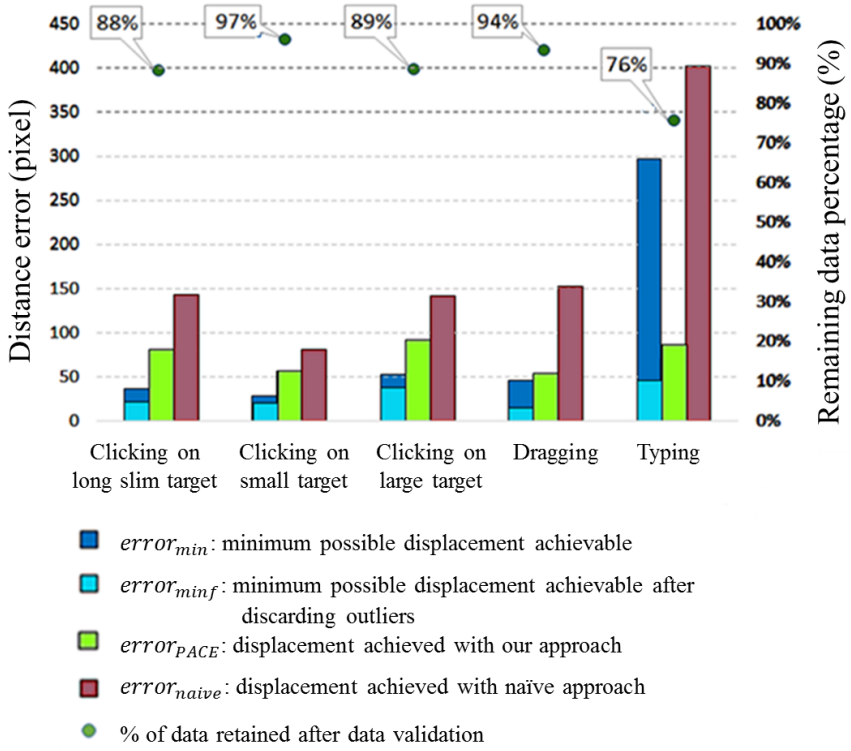


Figure 5-7. The displacement between gaze points identified with different approaches and location of corresponding interaction events.

In comparison, $error_{naive}$ (red) from the naïve approach is larger than $error_{PACE}$ (green) from our approach, across all interaction behaviors. The reason is obvious when one considers the U-shape pattern seen in most of the interaction behaviors (Figure 5-2), as the displacement falls to a minimum before the event, and then actually rises again just before the event. For example, with mouse drags, using the point at the event moment gives $error_{naive}$ of 153.3 pixels, while $error_{PACE}$, at 53 pixels, approaches the lowest possible $error_{min}$ (46 pixels). On average, $error_{naive}$ is 184.2 pixels while $error_{PACE}$ is 73.6 pixels.

It is also interesting to consider the amount of data that is retained after the two-layer validation process. We find that on average, 88.8% of the data is retained. The exception is typing, which has a low retention rate (76%), which is due to the large number of outliers. Incidentally, both the unfiltered $error_{min}$ and $error_{naive}$ are very large for typing interactions. However, our method is able to successfully identify these problematic data points, hence achieving a small $error_{PACE}$, which is close to that of mouse events. This is promising as keypress events are usually more numerous than mouse events, and hence it makes sense to find a means to include them as interaction-informed data.

Our results suggest that the proposed validation mechanism can effectively and precisely identify the reliable interaction-informed data, significantly outperforming the method based on the conventional assumption.

5.3.5 Evaluation in real-use contexts

To evaluate the accuracy and effectiveness of the PACE system, we recruited 10 subjects (university students, 6 female, aged 20-33) for a focused study. Subjects were asked to choose at least 3 of the following tasks for the data collection: browsing websites, coding in Visual Studio, writing in Notepad, creating a figure using Microsoft Paint, and playing a shooting game (the House of the Dead). These tasks were chosen to cover a diverse range of common user interaction activities and applications, and to contain diverse interaction types. For example, some of the tasks will involve relatively dense keypresses while others contain mainly mouse events, like clicking and dragging.

The experiments were run on an i7-2600MHz PC with 4GB RAM, a 22" monitor and a standard off-the-shelf webcam capable of recording at 30fps. Running PACE on this setup achieves a frame rate of 22fps and performing a model update with 1500 data points takes less than 500ms. The gold standard eye gaze position is measured by the Tobii EyeX tracker. Approximated by our face tracker, the head-to-monitor distance ranges from 372~892mm (M=663mm, SD=35.2mm); and head-to-camera pitch from -9.6º~56.6º (M=14.2º, SD=8.9º).

At least 1500 events were collected from each subject. Each mouse click and press of a letter key logs the gaze-point feature data from the preceding 3 seconds. Subjects were allowed to pause and continue the experiment as needed, even over multiple days if necessary. They were also free to adjust head pose, body posture, and chair position/height. The experiment lasted from 2 to 18 hours, depending on how long it

Table 5-3. Performance of our approach, compared with state-of-the-art appearance models that allow free head motion.

Methods	Error	Calibration	Data Required / Method Used
PACE	**2.56º**	**Implicit, Automatic**	**mouse/keyboard interactions**
Sugano et al.[103]	4º-5º	Implicit	click
Lu et al.[68]	2º-3º	Explicit	video
Lu et al.[67]	2º-3º	Explicit	image synthesis

took for the needed interaction events to be generated. On average, the subjects took around 4 hours to generate the required number of interaction points.

Table 5-3 shows the performance achieved by our method, as compared against similar state-of-the-art appearance-based methods that rely on webcam signals. The performance is measured as the average error over all collected instances and subjects. It is encouraging to see that our method achieves an average error of 30.9mm (*i.e* 2.56º visual error, calculated using the approach in Sugano et al [103]), which is comparable to state-of-the-art approaches that require explicit calibration. This is a promising result, given that PACE (1) does not require explicit calibration and will automatically update itself to account for changes in light and posture variance, even across multiple sessions spanning over multiple days, (2) uses conventional off-the-shelf equipment that is commonplace in work environments, and (3) is tested using real applications and activities.

Figure 5-8 shows a graphical example of the performance of our system during a browsing activity. The blue line shows the eye movement estimated by PACE, while the red line denotes the true trajectory, measured by the Tobii tracker. It is seen that the PACE eye positions closely approximate that from the Tobii for the majority of the time, without using any additional equipment.

It is informative to consider the improvement in performance as a number of data increases. We train two models using the same random forest algorithm. One is trained on data collected under the naïve assumption. The other is the PACE model, trained on

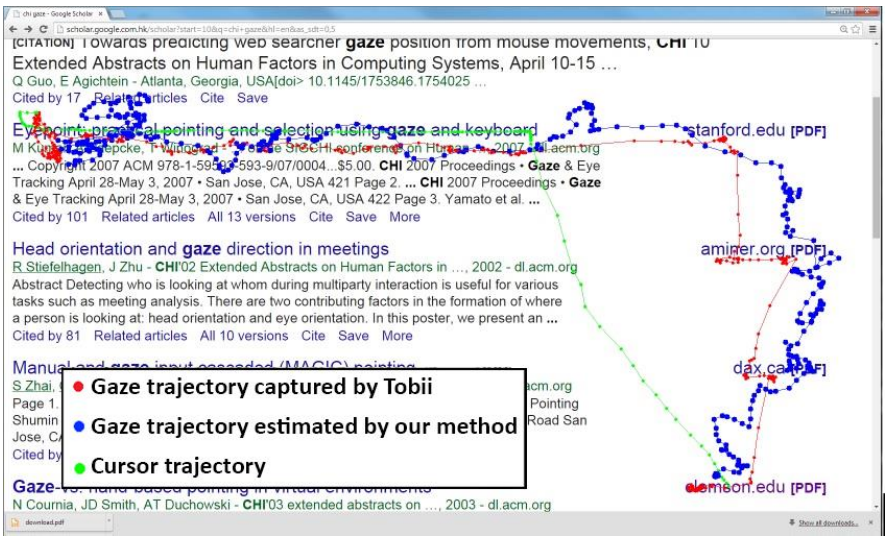


Figure 5-8. Trajectories of user gaze as estimated by PACE (blue) and as captured by the Tobii EyeX eye tracker (red). The cursor trajectory (green) is included for reference.

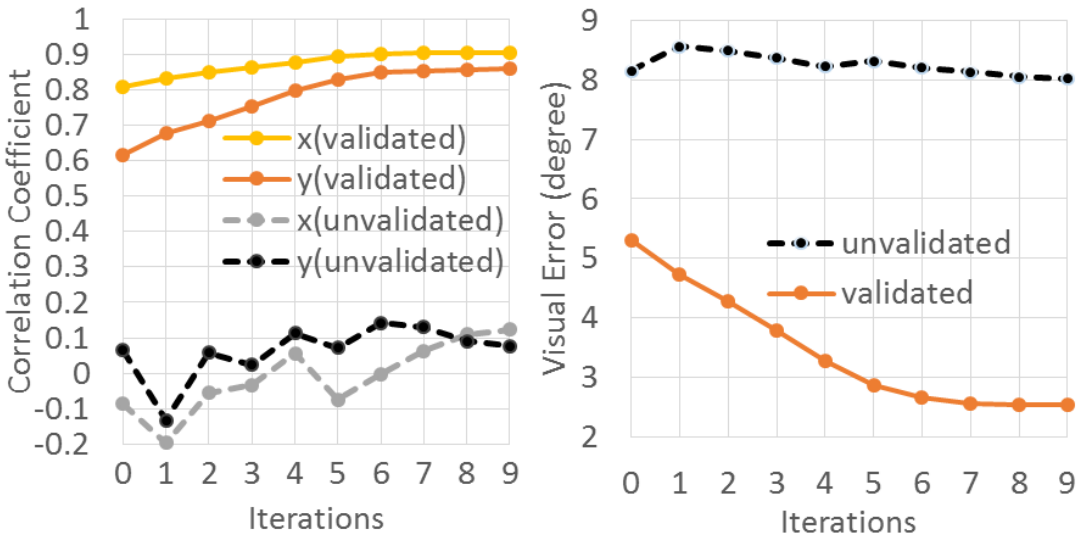


Figure 5-9. Comparison of PACE and naïve models. Change in performance (Correlation and Visual Error) as data increases. Each iteration consists of 150 interaction events.

data identified through the 2-step validation process. The models are updated and retrained every 150 interaction events (mouse clicks or keypresses). This means that the naïve model gets 150 new data instances per iteration, but the PACE model will get fewer instances, as the data-driven validation will invariably filter out some unreliable data points.

Figure 5-9 compares the performance (correlation and visual error) of the two models. Along the *x*-axis, each point represents one iteration of 150 interaction events. The performance of the naïve model fluctuates considerably – the visual error hovers around 8º, and the correlation never increases beyond 0.2. The performance of PACE, on the other hand, improves monotonically as additional training data is provided. The correlation reaches an impressive 0.90 and 0.85 for the *x*- and *y*-coordinates respectively. Regarding the visual error, there are growing divergence between the performance of the learning with unvalidated data and validated data. The visual error of PACE drops steadily to 2.56º. This is further evidence that shows that our validation mechanism is effective as well as necessary for collecting interaction-informed training data in real-use situations.

5.4 Summary

This chapter describes PACE, a Personalized, Auto-Calibrating Eye-tracker system that can be integrated into standard interactive computing systems. The assumptions behind PACE are informed by an in-depth study on the relationship between eye gaze and interaction location for several common types of interactive behaviors. Based on the results of this study, we develop a novel approach that automatically identifies the moment of best gaze-interaction alignment and a further data validation mechanism that accounts for user differences and context.

Experimental evaluations demonstrate that PACE can effectively extract good training data from daily interaction activities to build a reliable eye tracker with automated and implicit self-update capability. The performance thus achieved is comparable to those from similar state-of-the-art methods. However, PACE has the advantage that it automatically updates and re-calibrates itself and is therefore able to adjust to variances in conditions over multiple days and sessions.

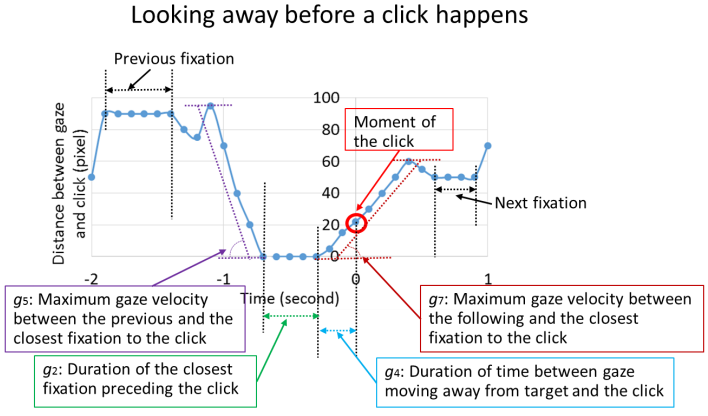
Chapter 6

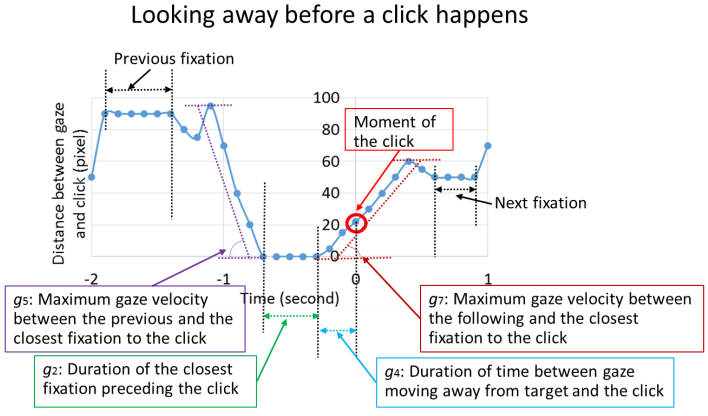
Sensing Stress from Gaze-Click Patterns

Selected notations and abbreviations used in this chapter

$\boldsymbol{e}$	set of eye features, including $\boldsymbol{e}_r, \boldsymbol{e}_l$ for the right and left eye, respectively
$\boldsymbol{F}$	gaze-point/ gaze-click feature vectors in a 3-second gaze-point/ gaze-click window
$\boldsymbol{f}_j$	the *j-th* column of $\boldsymbol{F}$, i.e. the sequential data of the *j-th* gaze-point/ gaze-click feature in the time window, $\boldsymbol{f}_j \in \Re^{m\times 1}$
$\hat{\boldsymbol{f}}_j$	filtered sequential data of the *j-th* gaze-point feature in the time window
$\boldsymbol{g}_c$	interaction point, i.e. location of cursor or caret
$\boldsymbol{g}_w$	estimated gaze point by the webcam data
g_j	the *j-th* element in the gaze-click feature vector
M	number of facial landmark, =66
m	number of frame in the gaze-point/ gaze-click window prior to an interaction event, =300
n	number of gaze features, =12 for gaze-point feature vector and =6 for gaze-click feature vector
t_p	time preceding an interaction event
$\boldsymbol{v}$	gaze-point feature vector, including the head and eye gaze features, $\boldsymbol{v} \in \Re^{n\times 1}$
$\tilde{\boldsymbol{v}}$	estimated gaze-point feature vector that corresponds to the aligned moment of the gaze the interaction cues, $\boldsymbol{g}_c$.
$\boldsymbol{x}$	locations of the facial landmarks in a particular frame
$\boldsymbol{x}_i$	the 3D coordinates of the *i-th* facial landmark
λ	distance threshold in the data-driven validation, =240 pixels in our case
ξ	fixation threshold for gaze-click features

Here are two illustrations of some of the gaze-click features ($g_2, \ldots, g_8$) from the 3-second window surrounding (during/ preceding) a mouse click:





The previous chapters investigate different techniques to facilitate the learning of personalized model through the adaptation of the generic knowledge, and intelligent use and acquisition of the personal data. This chapter exploits the cross-modal feature for the detection of the human affective state. The cross-modal feature describes the coordination of human behaviors from different modalities. They show promising results in affect detection and a certain degree of user-independency.

Specifically, we introduce a set of gaze-click features to describe the gaze movement behavior surrounding a mouse-click. The chapter first presents the description of our human study on solving math questions under different stress levels. It then describes the details of the gaze-click features, followed by the techniques we used to detection the mental stress. Experimental results show the effectiveness of our techniques and the generalizability of the proposed features for user-independent modeling. The results also suggest that it may be possible to detect stress in the wild in a non-intrusive fashion without the need for specialized equipment.

6.1 Constructing the StressClick Dataset

The first task in our study is to build a dataset that reliably captures human interactive behavior in stress and non-stress conditions under conditions that are comparable. Stress may well interleave with interest, attention, and workload. It is difficult to fully separate or clearly define such mental state(s), due to the complexity of human affect. Since we intend to investigate the relation between gaze-click pattern and stress, our experiment will have to be designed to eliminate other possible biases.

Prior studies show that recursive mental math calculation [1][73][104][111] and time pressure [58][112] are effective in inducing cognitive stress. We, therefore, select a math calculation task for evaluation.

Figure 6-1 shows the experimental interface. The upper- left and right parts display two automatically-generated math expressions. The user is asked to calculate the result of these two expressions, and determine which (if either) is greater. Pressing the “show” button (the green circle between the two equations) displays the answer buttons (“>”, “=” and “<”), which are randomly placed each time a new question is generated. This additional step ensures that the subject’s mouse movement (and gaze) will start from the “show” button and move towards one of the answer buttons. To investigate the influence of the click target attributes, the answer buttons can be varied (diameter: 70,

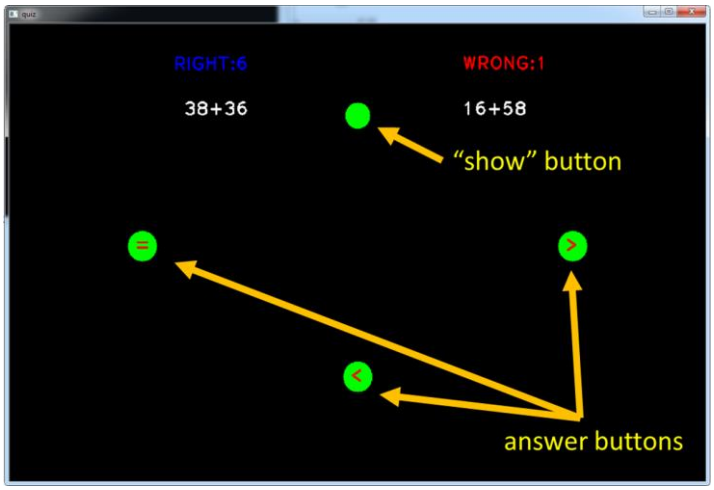


Figure 6-1. Experimental interface during a calm session.

120 pixels) as well as their distances to the "show" button (450, 700 pixels) between different experiment sessions.

The experimental setup is shown in Figure 6-2. The gaze data in the 2 seconds preceding and 1 second after a click on the answer buttons is logged for analysis.

The difficulty of the math expressions is varied to induce states of calm and stress. The calm session involves twenty 1-digit addition and subtraction questions. This is adjusted to 2-digit math for the stress session. To ensure the difficulty of the task, the numeric difference between the results of the two expressions is constrained to be no more than 10. To further induce stress, a countdown time bar is added to the bottom of the interface during the stress sessions. If the subject fails to answer within the allotted time, the interface advances automatically to the next question.

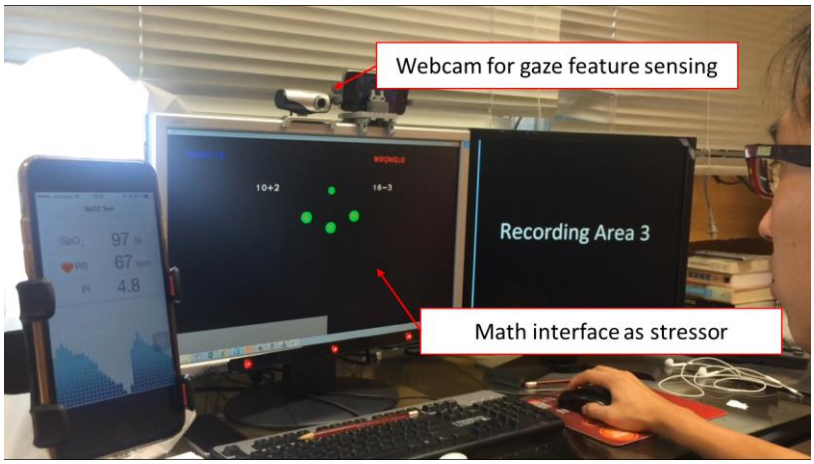


Figure 6-2. Experiment environment. A common webcam is used to sense the eye features. The math interface serving as the stressor is displayed in maximized mode.

To allow for differences in math aptitude and to familiarize subjects with the interface, the experiment starts with a short calibration session. Subjects are asked to complete a number of 2-digit questions as quickly as they can. The time spent on each question is recorded. The mean plus 2 standard deviations of the time taken to finish a question is taken as the time constraint for the stress induction session. This is decided based on our pilot study of 3 subjects on solving 500 hundred math calculation questions. Assuming that speed can be well approximated by a Gaussian distribution, this ensures that all subjects should have just enough time to finish most of the questions while still ensuring a somewhat stressful experience. The use of Gaussian distribution is because that we do not have a clear idea of what this probability should look like for the general subjects. Therefore, Gaussian distribution can be a good approximation under this circumstance, as in other engineering issues Gaussian distribution is used a default distribution to solve the problem.

The design of our experiment involves three factors (stress level, size, and distance of the click target), with two levels of each factor. Our experiment therefore contains 2×2×2 sessions, each of which has a different factor combination, with a randomly generated section order. This introduction of size and distance variations of the target is due to the impact of the effects of Fitt's law on the gaze-click patterns. We intend to study the how this effect compared with the mental stress influences the gaze-click patterns. At the end of each session, the subject is asked to report his/her level of stress on a 9-point scale, after which he/she is asked to take a break and listen to peaceful music to calm down before beginning the next session.

We recruited 20 subjects (13 males, aged 20-33) for our study. A standard off-the-shelf webcam is used to capture the visual signal (resolution 640×480; 30fps) and a 22" monitor at 1680×1050 resolution displays the math interface in full-screen mode. Removing the questions that the subjects fail to answer (<5%) in time gives us a total of 3818 click points over all subjects.

Inspecting the data, we note that the self-reported stress level (on a 9-point scale) has a marked difference between the 1-digit section (M=2.10, SD=1.09) and the 2-digit section (M=7.28, SD=0.94). This difference confirms that the experiments are successful at inducing stress during the appropriate sessions.

6.2 Extracting Gaze-Click Pattern

Our method uses a standard webcam to capture video of the user's head and shoulders. To ensure that we can accurately deduce the gaze behavior of the subject from this video, we use two tracking models to identify the face and eye landmarks from the frames of the video stream, respectively. These landmarks are then piped as input to a two-layer feature extraction mechanism. The first layer continuously extracts six *eye features* from the changes of the eye-related landmarks. The second layer is triggered by each mouse-click event, whereupon it extracts eight *gaze-click features* based on the eye feature signals in the 3-second time window surrounding the click.

6.2.1 Extracting six eye features

This section introduces the six eye features that describe the essential eye geometry of each frame, including the openness, horizontal and vertical rotation of both eyes (see Figure 6-3).

These eye features are measured according to the facial and eye landmarks. To obtain accurate locations of the facial landmarks, we employ the Supervised Descent Method (SDM) [118] to track 48 facial landmarks. Since the SDM model tracks landmarks based on both textures of the landmarks' local patches and their geometry inter-dependency, it performs robustly against illumination influences and gives reliable landmarks that describes the eye contour (see the red landmarks in Figure 6-3).

In order to obtain an accurate description of eye geometry, it is crucial that the pupil center is properly located. However, in unconstrained situations, the appearance information of the eye region often has no clear cue of the pupil center. Additionally, we observe that in real-use scenarios, low video resolution and reflections on glasses and cornea usually makes the region of the pupil and its periphery unobservable. As a result, simple techniques based on image texture or edge often fail to reliably track the pupil center.

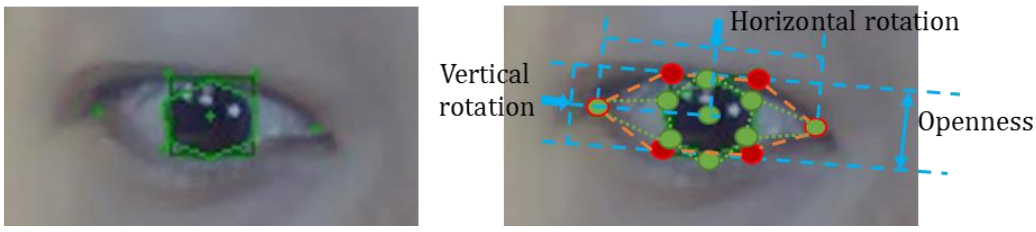


Figure 6-3. The eye landmarks are identified and tracked from the webcam image.

To mitigate the localization issues caused by the unconstrained environment and noisy images with low resolution, we use eye geometry to estimate the pupil center. The general idea is to track landmarks with good salient features on the iris contour and eyelid corners, and deduce the pupil center locations based on their geometric interdependency. We, therefore, apply eye CLMs [96] to track the eye landmarks, including the pupil center. Unlike the 28-landmark model in [115], our eye models use only 9 landmarks, as shown in Figure 6-3. Given the low resolution and blurred image in real-use, we believe only those fiducial points with the most salient features can facilitate tracking. Besides, reducing the number of tracking landmarks can decrease the model complexity, which is highly favorable for real-time systems

We manually annotated 3000 eye images from 5 people (not from the StressClick subjects), and used this data to train the eye tracking model. It is surprising that the pupil center patch learned from our annotated data shows an edge-like pattern, which may be caused by reflections. This supports our hypothesis that additional landmarks inside the iris region may not help the reliable tracking.

To validate the eye tracking state, we apply a robust but rough tracking model that uses the integral image [83] to approximate the pupil center from the eye image, with the assumption that the darkest rectangular area is the iris. If the CLM estimation is too distant (>half of the distance between the upper and lower eyelids) from the integral image approximation, we assume that the CLM tracking has failed and we redetect the eye for tracking.

Based on the landmarks identified by the face and the eye trackers, we extract eye features $\boldsymbol{e}_r, \boldsymbol{e}_l$ from the right and left eye respectively. Each $\boldsymbol{e}$ is calculated according to the eye landmark distances.

$$\boldsymbol{e} = \langle \alpha(d_5 + d_6), \frac{d_1}{d_1 + d_2}, \frac{d_3}{d_3 + d_4} \rangle \tag{6.1}$$

where α is a constant scale factor that bounds the value of the eye openness feature range within [0,1).

6.2.2 Identifying eight gaze-click features

To understand the gaze movement pattern, which can be indicative of mental stress and in the meanwhile observable from the webcam signal, we propose eight potentially useful gaze-click features. These features describe the type, duration and velocity of the

gaze movements within a *gaze-click window* 2 seconds preceding and 1 second after a click event.

We calculate the gaze-click features based on the temporal changes of the eye features $\boldsymbol{e}_r$ and $\boldsymbol{e}_l$ within the gaze-click window. The gaze-click features are all defined with respect to a mouse click and describe the gaze pattern preceding (g_4, g_5, g_6), during (g_1, g_2), and after (g_3, g_7, g_8) the click event. Table 6-1 shows the description of each gaze-click feature and the implied mental state.

Figure 6-4 and Figure 6-5 illustrates the gaze-click features describing temporal durations within the 3-second window surrounding a mouse click. The x-axis indicates the time prior to (negative) and after (positive) the mouse click. The y-axis indicates the distance between the actual gaze point and the location of the click. A distance of zero indicates overlap between the eye gaze point and the click location. Similarly, a positive distance means a fixation somewhere else. Since most of our features investigate the fixations and non-fixations surrounding a click, it is worth noting that in

Table 6-1. Description and mental state implication of gaze-click patterns extracted from the eye features and used in this work. All features are calculated relative to a given mouse click.

Index	**Feature description**	**Mental state implication**
g_1	Existence of a fixation in the 0.5s period preceding a click event	Level of visual attention to click target
g_2	Duration of the closest fixation preceding/during a click event	
g_3	Reaction Latency - duration of time in which eye remains fixated after the corresponding mouse click	Reaction latency of moving to the next task
g_4	Click Latency - duration of time between gaze moving away from target and corresponding mouse click.	"Hastiness" of the user in locating the next target.
g_5	Maximum gaze velocity between the previous and the closest fixation to a click event	"Hastiness" of the user in locating the current target
g_6	Duration between the previous and the closest fixation to a click event	
g_7	Maximum gaze velocity between the following and the closest fixation to a click event	"Hastiness" of the user in locating the next target.
g_8	Duration between the following and the closest fixation to a click event	

the absolute majority (95%) of the cases in our dataset, there is indeed a fixation during (Figure 6-4) or shortly preceding a click (Figure 6-5).

Figure 6-6 shows some annotated examples of gaze-click patterns under both calm and stressed conditions. Some of the gaze-click pattern features are magnified for illustration (in blue frames). It is interesting to note that under stressed conditions, the user's gaze often shifts away from the fixation point *before* he/she clicks on it

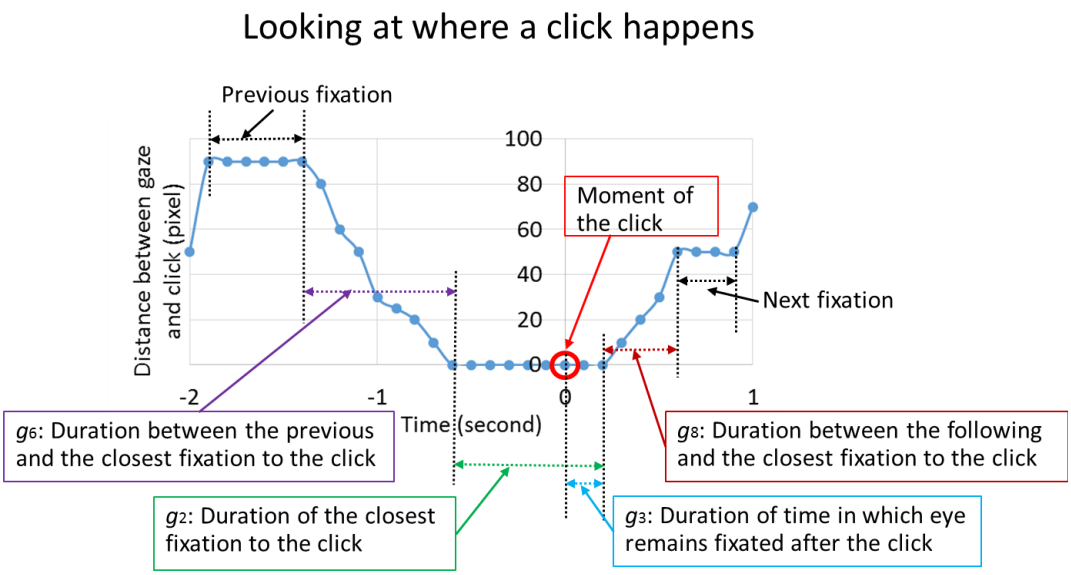


Figure 6-4. Illustration of gaze-click features from the 3-second window surrounding a mouse click during the user is looking at where a click happens.

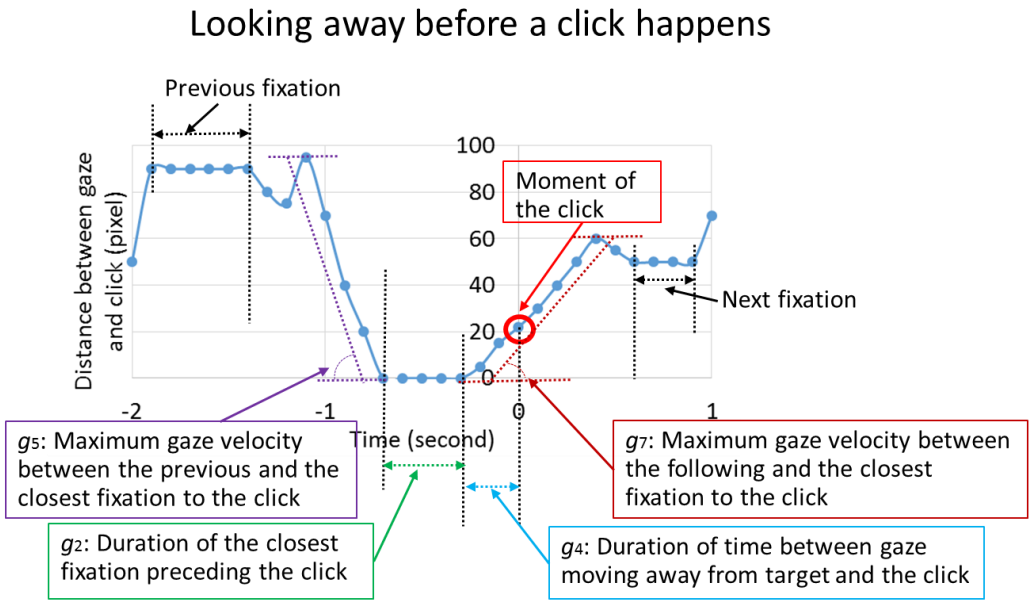


Figure 6-5. Illustration of gaze-click features from the 3-second window surrounding a mouse click during the user is looking away before a click happens.

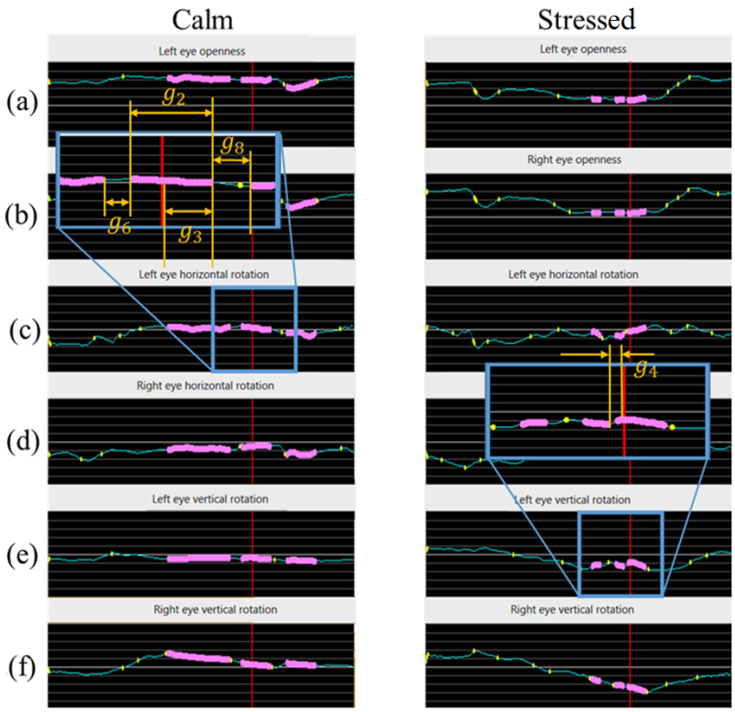


Figure 6-6. Example eye features from the 3-second window surrounding a mouse click during calm (left) and stressed (right) conditions. The red line indicates the moment of the click, the 3 fixations nearest to the click are highlighted in pink, and the yellow moments are those with large signal change.

("stressed" column, e.g. Figure 6-6, 5th row**)**. To accommodate this phenomenon, we define the fixation corresponding to a click as the fixation that starts at least 80ms (a minimal fixation duration [39]) before the click occurs. Only in cases where such a fixation does not exist is the corresponding fixation defined as the closest fixation immediately preceding the click.

To compensate for visual noise in the raw eye signals and the uneven sampling rate, we first resample the eye feature temporal sequences by interpolating them to 100Hz. We then apply band-pass filtering to remove the unnatural high-frequency noise. Given the eye feature signal $\boldsymbol{F} \in \Re^{m \times n}$, where m(=300) is the number of data points in a 3-second gaze-click window and n(=6) the number of eye-related features, we identify the fixation periods using the dispersion-based approach [93]. All the signals in a fixation period (FP) (>80ms) should be subject to the condition:

$$\max(\boldsymbol{f}_i^{FP}) - \min(\boldsymbol{f}_i^{FP}) < \xi \tag{6.2}$$

where $\boldsymbol{f}_i^{FP}$ represents the *i-th* eye feature in a fixation period and ξ is the fixation threshold.

Table 6-2. Correlations between the gaze-click features and stress level, click distance and target size.

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
Calm/Stress	-0.04	-0.35	-0.26	0.09	0.16	0.14	0.03	0.04
Click distance	-0.06	-0.05	-0.05	0.12	0.04	0.05	0.03	0.08
Click target size	0.01	-0.06	-0.03	-0.01	0.09	0.03	0.03	0.01

To quantify the degree of "hastiness" on the part of the user, we calculate the *maximum gaze velocity* between two fixations as the sum of the maximum slope of each eye signal during the non-fixation period (NFP) as

$$\beta \sum_{i=1}^{n} (\max(\boldsymbol{f}_i^{NFP}) - \min(\boldsymbol{f}_i^{NFP}))/\Delta t \tag{6.3}$$

where Δt is the time difference between the two data points at the maximum and minimum moments. β is a scale factor which makes the value of this velocity feature comparable with the gaze-click features depicting temporal duration – the values of a clear majority of the gaze-click feature range in [0,1).

6.3 Understanding Gaze-Click Behavior under Calm and Stressed Conditions

Replacing the calm and stress labels with numeric "0" and "1" to represent the stress level, allows us to calculate the correlation between the gaze-click features and the experiment independent factors. Table 6-2 gives the correlations between the gaze-click features and the stress level, click distance (between "show" and "answer" buttons) and target size (i.e. "answer" button diameter).

It is not difficult to see that the gaze-click features basically have no clear correlation with the click target distance and size. It therefore indicates that the on-screen distance between the locations of the cursor and the click target does not have a significant impact on the gaze movement behaviors surrounding a mouse-click. Nor does the size of the click target. However, it is interesting that two gaze-click features, the closest fixation duration preceding/during a click and the reaction latency after a click, show fair correlation with calm/stress, compared to click distance and target size. The negative correlation values suggest that long fixation duration and reaction latency associate more strongly with the calm state than stress state, which appears to make sense because subjects who are not under stress may operate less hastily and more slowly.

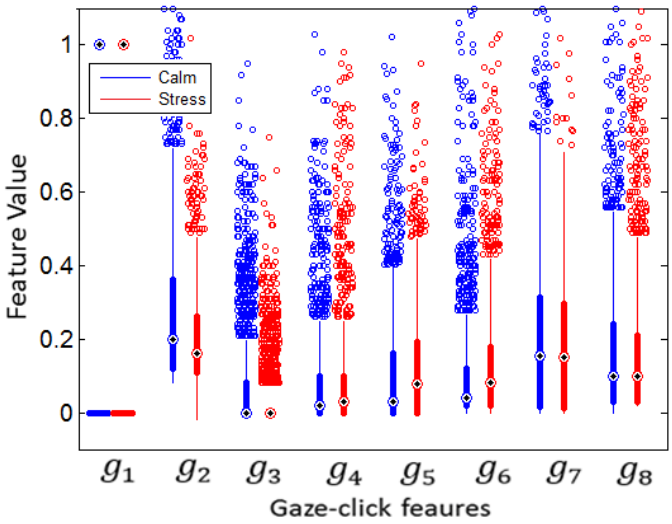


Figure 6-7. Box plot of the gaze-click features under the calm and stressed conditions. The central mark is the median, the thick bar covers the 25th and 75th percentiles and the thin line extends to the most extreme data points not considered outliers, and outliers are plotted individually in circles.

Figure 6-7 compares the overall values of the individual gaze-click feature under calm (blue) and stressed (red) conditions. The dotted circles mark the median, the thick bar covers the 25th and 75th percentiles, the thin line extends to the most extreme data points not considered outliers (defined as values >*q3*+1.5(*q3*-*q1*) or <*q1*-1.5(*q3*-*q1*), where *q1* and *q3* are the 25th and 75th percentiles), and outliers are plotted individually in circles. It is not difficult to see that some gaze-click features differ considerably between calm and stress conditions. For example, the average duration of the corresponding fixation, g_2, in stress (M=0.20) is 30.3% shorter than that of in calm contexts (M=0.29). Similarly, the reaction latency, g_3, in stress (M=0.04) is 46.7% lower than that of in calm (M=0.07). In contrast, the click latency g_4 (calm M=0.08; stress M=0.11), velocity g_5 (calm M=0.10; stress M=0.12) and duration g_6 (calm M=0.11; stress M=0.14) of the preceding non-fixation period in stress are 31.4%, 14.5% and 21.1% higher than those of in calm conditions.

Intuitively, these differences make sense. It is highly likely that subjects would fixate longer (g_2, g_3) on the target when calm than when stressed. Likewise it is common that stress may induce people to move their gaze away (g_4) from the target, scan more hastily and at higher speed (g_5) and scan, rather than read, more often (g_6).

To further explore the probability distribution behind the overall differences, we present the conditional probability mass functions (pmf) of each gaze-click feature given both calm and stress mental conditions in Figure 6-8. For the time-related features

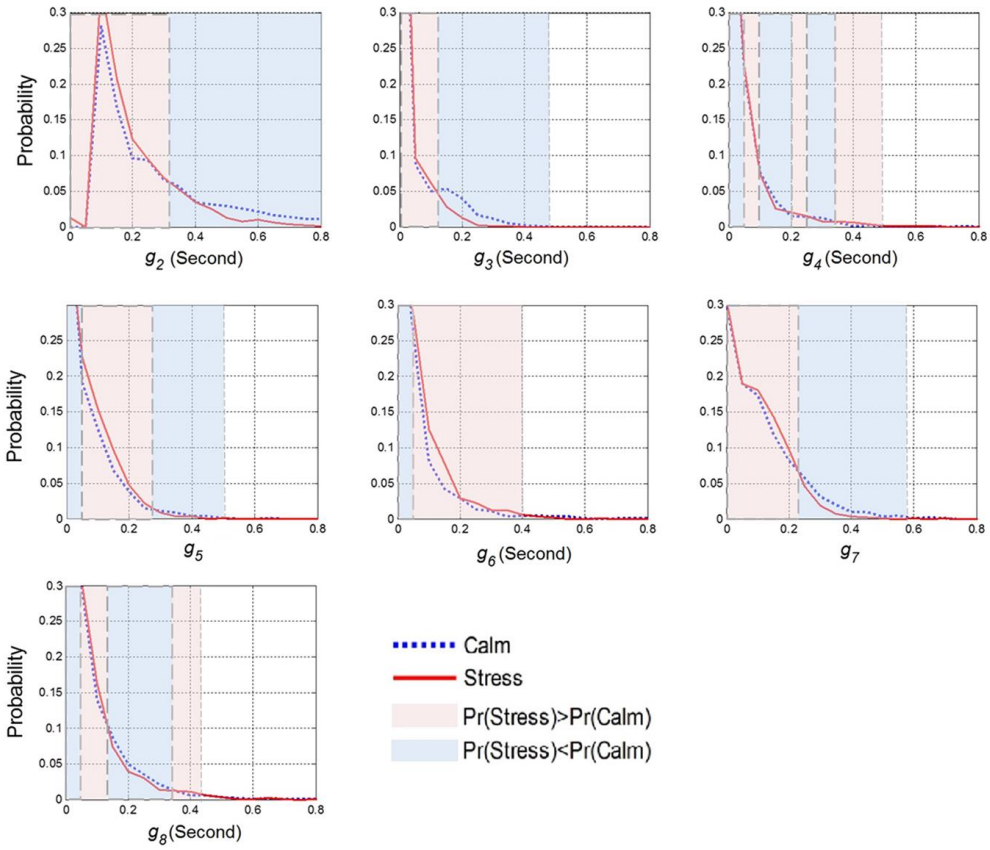


Figure 6-8. Conditional probability mass function of the gaze-click features. The red line indicates stress, and the blue dotted line indicates the calm condition. Light red shadow denotes the stress probability is higher than that of calm, and the light blue shadow shows that calm probability is higher than that of stress. Some gaze-click features ($\boldsymbol{g_2, g_3, g_5, g_6, g_7}$) present clear pattern under different stress levels, while others' patterns ($\boldsymbol{g_4, g_8}$) are rather ambiguous in term of conditional probability distribution.

(g_2, g_3, g_4, g_6, g_8), each point on the curve represents the probability of feature values in the corresponding 50ms bin; with respect to the velocity-related features (g_5, g_7), each bin represents an incremental value of 0.05. The red line indicates the probability of stress and the blue dotted line indicates calm. For a clear demonstration, where the probability of stress is higher than that of calm, i.e. $\Pr(\text{Stress}) > \Pr(\text{Calm})$, we annotate the region in light red, and light blue for the converse situation, i.e. $\Pr(\text{Stress}) < \Pr(\text{Calm})$. We observe that an absolute majority of probabilities drop to zero when the value of the feature goes over 0.8. Therefore, we show feature distributions in the value range (x-axis) of [0,0.8]. Similarly, for display purposes, we show the probability range (y-axis) of [0,0.3]. It should be noted that we follow the definition in [39] that a fixation should last at least 80ms, so the figure of g_2 has no value for durations shorter than this threshold.

We have not shown g_1 in the figure, since it is a binary feature, and the probability difference of a closely preceding fixation in calm (95.2%) and stress (96.6%) is marginal. Encouragingly, most of the gaze-click features (g_2, g_3, g_5, g_6, g_7) present clear patterns under different stress levels. In other words, the relationship between the value of these features and stress level are relatively stable and consistent. However, other features' (g_4, g_8) relations with mental stress are vague. As their values increase, the higher probability switches back and forth between stress and calm.

Generally speaking, the data indicates that the gaze behaviors surrounding a mouse-click event show a certain degree of connection with the changes of mental stress. Although the connection appears slightly ambiguous between an individual gaze-click feature and the human stress level, it can be indicative for discriminating stress and calm given a proper model exploring the relation with multiple features.

6.4 Evaluating Stress Detection at Individual Click-Level

To model stress from the eight gaze-click features, we adopt the random forest algorithm [8], which is an ensemble learning method that has been shown to achieve good performance in various applications. The parameters are empirically determined based on grid searches using 10-fold cross-validation on the training set. Our final model contains 75 trees with max depth truncated to 4, and each tree employs 3 features for random selection.

In order to fully investigate the generalizability of the proposed features, we present the results of both within-subjects and between-subjects studies. For the within-subjects evaluation, we build a user-dependent model for each individual subject and use 10-fold cross-validation for evaluation. The final performance is the average performance

Table 6-3. Performance comparison of click-level detection between user-dependent and independent models

	User-Dependent Model	User-Independent Model
CCR	65.72 (51.05)	60.27 (51.1)
F1	0.66 (0.36)	0.60 (0.35)
AUC	0.70 (0.5)	0.63 (0.5)

Numbers in parentheses denote the baseline performances. The performance is the weighted average results across different subjects, whose values may have slight difference from the average, due to the differences of data amount of subjects.

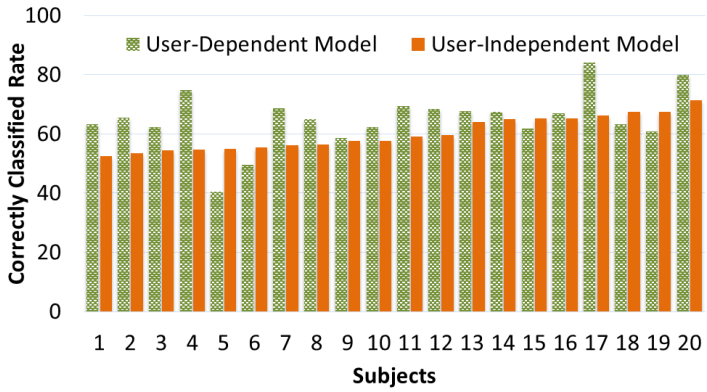


Figure 6-9. Classifying stress at individual click-level with user- dependent and independent models. User-dependent evaluation uses 10-fold cross-validation on individual subject's data and averages over all folds. User-independent evaluation reports results on leave-one-subject-out cross-validation. The subject IDs are ranked by the ascendant performance of the user-independent model.

across all folds. The between-subjects study employs leave-one-subject-out cross-validation, testing on each subject in turn. For reference, we provide the baseline performance given by a naïve classifier that predicts the majority class in the training set.

Table 6-3 summarizes the performance comparisons of both user- dependent and independent models to the baselines. The weighted average of correctly classified rate (CCR), F1-measure, and area under the receiver operating characteristic curve (AUC) across subjects are used as performance metrics. The comparison shows that our models significantly outperform the naïve classifiers. The user-dependent model (F1=0.66), as expected, outperforms the user-independent model (F1=0.60). This makes sense, as the user-independent model needs to accommodate differences between individuals, which generally can be difficult. Our technique gives satisfactory results despite being agnostic to the click target size and distance and inferring based on only individual clicks.

For a thorough understanding of the model performance on every individual, Figure 6-9 presents the CCR comparison across subjects. Performances of the user-dependent model are marked in the green and user-independent model in orange. The subject IDs are ranked in ascending order by the performance of the user-independent model. Although the user-independent model (M=60.2, SD=5.6) has a modest 4.8% reduction in performance on average compared to its counterpart (M=65.0, SD=9.2), it is encouraging that it performs more stably with a smaller standard deviation. In addition, the user-independent model actually succeeds in outperforming the user-dependent

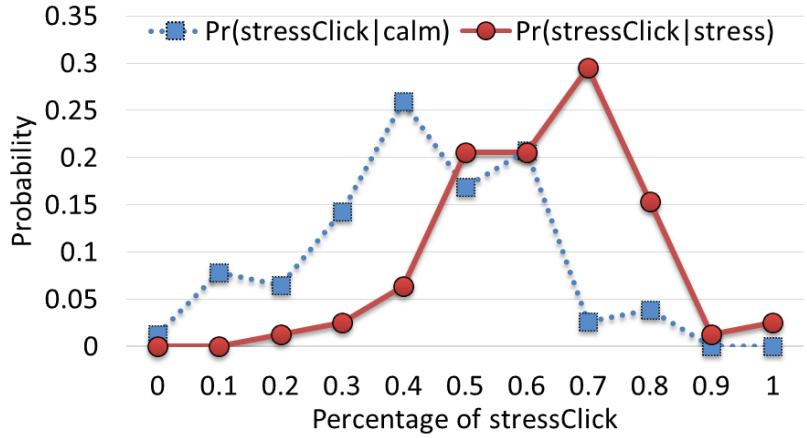


Figure 6-10. Conditional probability functions of StressClick given the subject is in calm and stressed conditions. People in stress tend to have a higher percentage of StressClick.

model in a quarter of our cases (5 out of 20 subjects). This result suggests the generalizability of our proposed gaze-click features - in other words, these features are general enough to adapt to different individuals.

6.5 Evaluating Stress Detection at Session-Level

Normally, people would stay in a similar mental state during the period of conducting a task. Inferring stress from the behaviors surrounding only one click may be influenced by the randomness of human behaviors. Given the success of stress detection based on the gaze-click pattern of an individual mouse event, we believe it makes sense to consider the sequential mouse-click events for stress detection.

We define the mouse-click whose gaze-click pattern leads to a stress prediction by the click-level classifier (i.e. random forest) as *StressClick*. Figure 6-10 presents the conditional probability of the percentage of StressClick in the stress and calm sessions.

It is not difficult to see that the conditional probability of StressClick in the stress session peaks at 0.7, which means there is a clear chance that StressClick occupies the dominant percentage of clicks in a session. In contrast, the conditional probability of StressClick in the calm sessions is likely to be lower than in stress. This means that the percentage of StressClick events can be useful for session-level stress detection.

We introduce a 2^{nd}-layer classifier to recognize stress given the click-level predictions. We construct 3 features for the 2^{nd}-layer classifier: (1) number of StressClicks, (2) number of clicks being considered, and (3) the ratio of StressClicks to the total number of clicks. The rationale of the first feature is that it may take a certain number of StressClicks to confirm a reliable stress prediction. The third feature, a

Table 6-4. Performance comparison of session-level detection between user-dependent and independent models

	User-Dependent Model	User-Independent Model
CCR	74.0 (49.35)	80.5 (50.0)
F1	0.74 (0.33)	0.79 (0.33)
AUC	0.73 (0.5)	0.89 (0.5)

Numbers in parentheses denote the baseline performances.

seemingly redundant description of the first two, explicitly instructs the classifier to apply the ratio of StressClick for a time period. Given the simplicity of the 2nd-layer features, a logistic classifier [11] is used for stress detection from multiple clicks.

Table 6-4 summarizes the performances of the user- dependent and independent models for the session-level prediction. It is very encouraging that the session-level user-independent model (F1=0.79) outperforms the user-dependent model (F1=0.74), given enough click data. Furthermore, as expected, session-level user- dependent and independent models outperform their click-level counterparts, with 8.31% and 20.23% CCR improvements, respectively. The F1-measure of the user-independent model also improves by around 0.2 from 0.6 to 0.79. This also supports the validity of the ratio feature.

Figure 6-11 further shows the performance on individual subjects. The session-level user-independent model (M=79.4, SD=16.9) outperforms the user-dependent model (M=71.3, SD=25.3) in both overall accuracy and stability (with smaller SD). This is

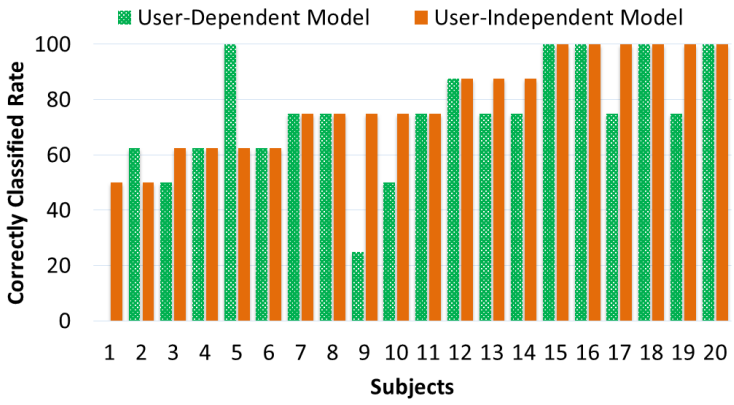


Figure 6-11. Classifying stress at session-level with user- dependent and independent models. The subject IDs are ranked in ascending performance of user-independent model.

probably because there is not sufficient data to build an accurate user-dependent model at the session level. However, when more training data is available, the session-level user-independent model performs quite well. It achieves over 80% accuracy on 45% of our subjects.

Comparing to previous state-of-the-art work that uses off-the-shelf devices for non-intrusive stress detection, Sun et al [104] reports 71% accuracy based on 30 samples of mouse movements in the user-dependent evaluation. Our method compares favorably with theirs and maintains this performance even in the user-independent paradigm.

6.6 Summary

This chapter presents a technique that aims to non-intrusively detect user stress through the gaze-click pattern. Using a series of multi-user experiments, we empirically demonstrate the impact of stress on the gaze-click pattern, which has been largely ignored in previous work. We also propose the cross-modal gaze-click features for stress recognition and investigate their effectiveness in both user- dependent and independent studies. Our results show that not only is it feasible to detect user stress through non-intrusively collected data, but also that our proposed features are generalizable across different users.

Given that our method does not rely on gaze point estimation, we believe that this is a significant contribution to affective computing and human-computer interaction, as it establishes a novel and reliable cross-modal approach to detect stress in a non-intrusive manner and thus can be used in the wild.

It is not difficult to see that our technique can also be applied in conjunction with mouse stiffness modeling [104] or take into consideration multiple clicks or click-trajectory history to facilitate stress classification. Our future work will therefore investigate more sophisticated multimodal signals with which to reinforce the stress-sensing model.

Chapter 7

Conclusion and Future Work

Although especial sensing equipment can provide some unique information, which can be indicative of the human state perception, like the thermal image, 3D facial mesh deformation, pupil dilation or even brainwave signal, the proposed techniques in this thesis rely on only the common computer equipment, such as webcam, keyboard, and mouse. This is because our motivation is to capture and understand the daily human-computer interaction behaviors in a low cost and non-intrusive manner. There are two reasons that motivate this approach: that further investigations can be widely done in the wild without being constrained by the laboratory setting, and to really facilitate the widespread applications in this field and contribute to the community as well as to the end user.

We present user-specific and user-adaptive approaches for facial affect recognition to address the model generalizability and the practical annotation issues. We developed techniques to implicitly collect and refine the interaction-informed data for gaze learning to facilitate visual attention analysis. We also investigated the feasibility of stress detection through the gaze-click patterns. This thesis concludes with a summary of the contributions and potential future work.

7.1 Contributions

Our main contributions are as follows:

- User-specific facial affect model:

We (1) propose a novel adaptive clustering approach to encoding facial response data from individual users; (2) devise a novel AMIL method that automatically identifies facial gestures from spontaneous expressions and associates them with human affects, given only segment-level labels; (3) collect a spontaneous facial response dataset; and (4) show the effectiveness of our method in modeling user-specific, spontaneous facial affects and demonstrate its superiority compared to its user-independent counterparts.

- User-adaptive facial affect model:

We (1) propose an efficient bootstrapping-based technique to transfer the generic knowledge of facial affects; (2) devise an individual data alignment technique to normalize data across individuals; (3) develop a simple but effective method to

aggregate the segment-level feature for multiple-instance learning and (4) present state-of-the-art spontaneous facial affect classification performances on four public datasets.

- Implicit gaze learning from interaction-informed data:

We (1) conduct an in-depth experimental study to investigate and quantify the gaze-interaction consistency across different behaviors; (2) propose an unobtrusive, adaptive, interaction-informed method that identifies the gaze-interaction alignment in daily computer use; and (3) demonstrate the effectiveness of our approach in multi-person evaluations across diverse interactive tasks.

- Sensing stress from gaze-click patterns:

We (1) present an in-depth behavior analysis that corroborates our hypothesis that stress affects the gaze-click pattern. This impact of stress has been neglected in previous gaze-cursor studies, and may be able to explain the conflicting results from prior human subject empirical studies. We (2) develop cross-modal features that describe the gaze-click pattern and exploit them for stress detection in a non-intrusive manner. This approach is promising for automated stress detection in the wild.

7.2 Limitations

This thesis investigates techniques for personalized model building. The experimental results are promising, however, there are still some limitations of the related studies.

The emotion elicitation in our facial affect analysis can be improved. Our current evaluation results indicate that the elicitation of some emotions is difficult, such as sadness. Since the effect of elicitation highly depends on the personal experience, a pre-experiment questionnaire can provide a better understanding of the participants, which may help to customize the elicitation materials for a more effective elicitation during the experiment phase. Furthermore, we used video clips from horror movies to induce fear, however, according to the post-experiment interviews we knew that these clips also induced a certain degree of disgust. It is interesting to investigate the mixture of feeling in the future. However, the mixed emotion data may be a constraint of the recognition performance for the study of AMIL in learning the personalized model. An improvement of the elicitation design may be needed in future studies.

The methods used in this thesis can be continuously studied and improved. Although the proposed AMIL technique shows encouraging results for categorical emotion

learning, it currently cannot deal with the dimensional affect recognition problems. It would be interesting to improve this technique to solve the regression issue. In addition, we used support vector machines and random forest in our studies. Given that the deep learning techniques give sweeping performance gains in many recognition problems, it should be of interest to benchmark with these techniques. Specifically, the long short-term memory neural networks (LSTM NN) could be a good choice for affect recognition [80]. However, to use the LSTM NN, the length of the video segments in the MIL facial affect problem may require special constraints according to the design of the LSTM NN model.

Another limitation is the computational issue from the system design. Our facial affect analysis method relies on the clustering technique to encode the facial expressions. Currently, the clustering of facial video frames is computationally expensive. It is of great interest to simplify this procedure, in order to make the model learning and update possible in real-time.

The limitation of the current evaluation metric is also worth noting. The studies in this thesis mainly use F1 score, correctly classified rate, precision, and recall for evaluation, following the traditional pattern recognition research protocol. However, for studies on personalized models, the amount of data that would be required from the target user is of interest. For example, a good learning technique for the personalized model should be able to obtain high recognition accuracy with as little target data or as little data annotation effort from the target user as possible. An ideal performance metric should include this information. A new performance metric should be designed for this purpose.

7.3 Future Work

7.3.1 Recognition of mixed emotions

This thesis focuses on the single-label classification problem for facial affect studies. We foresee, however, that a mixture of affects may occur in real-use scenarios. Apart from introducing a label that combines multiple affects, a potential solution is to just use all reported affects as valid bag labels when calculating the RFIAF values in the learning phase, and adopt a threshold or use a classifier to determine the occurrence of

each affect. Additionally, it will be also interesting to investigate the high-level mental state, such as “frustration” and “thinking” in the mixed emotion context.

7.3.2 Optimal adaptation

Although fast-PADMA presents encouraging empirical results, further studies can be conducted to investigate the aggregation of weak generic classifiers. Due to the practical need of constraining computational cost and the freedom from data release, our method currently suggests the same number of weak generic classifiers as the number of source subjects. We believe there could be diverse possibilities for ensemble learning. And it would be interesting to investigate the relation between computational cost and performance improvement.

7.3.3 Visual attention and mental state

PACE studies the technique to allow gaze learning based on webcam and normal interaction data. It makes possible the comprehensive eye gaze analysis *in-situ*. To further understand the human mental state, valuable future work includes the exploration of the relation between human affect and visual attention/ gaze movement pattern/ gaze-hand coordination.

StressClick also discusses the stress detection by the gaze-click pattern. It is not difficult to see that this technique can also be applied in conjunction with mouse stiffness modeling [104] or with multiple clicks or click-trajectory history to facilitate stress classification. Our future work will therefore investigate more sophisticated multimodal signals with which to reinforce the stress-sensing model.

7.4 Other Relevant Contributions

In addition to the main contributions previously described, the following describes some relevant contributions arising from my thesis project:

7.4.1 MelodicBrush

MelodicBrush is a novel system that connects two ancient art forms: Chinese ink-brush calligraphy and Chinese music. Our system uses vision-based techniques to create a digitized ink-brush calligraphic writing surface with enhanced interaction functionalities. The music generation combines cross-modal stroke-note mapping and

statistical language modeling techniques into a hybrid model that generates music as a real-time, auditory response and feedback to the user's calligraphic strokes. This system is, in fact, a new cross-modal musical system that endows the ancient art of calligraphy writing with a novel auditory representation to provide the users with a natural and novel artistic experience. The system is described in the following publications:

Huang, Michael Xuelin, Will W. W. Tang, Kenneth W. K. Lo, C. K. Lau, Grace Ngai, and Stephen Chan. 2012. "MelodicBrush: a novel system for cross-modal digital art creation linking calligraphy and music." In *Proceedings of the Designing Interactive Systems Conference on DIS '12*, 418-427.

Huang, Michael Xuelin, Will Tang, Kenneth W.K. Lo, C.K. Lau, Grace Ngai, and Stephen Chan. 2012. "MelodicBrush: a cross-modal link between ancient and digital art forms." In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts - CHI EA '12*, New York, New York, USA: ACM Press, 995.

Figure 7-1. The example artworks (above) and the user testing MelodicBrush (below).

7.4.2 Emotar

Watching movies has always been a popular mode of socialization and video sharing is increasingly viewed as an effective way to facilitate communication of feelings and affects. We develop an asynchronous video-sharing platform that uses Emotars, or avatars that indicate emotions, to facilitate affect sharing in order to create and enhance the sense of togetherness through the experience of asynchronous movie watching. We investigate its potential impact and benefits, including a better viewing experience,

supporting relationships, and strengthening engagement, connectedness and emotion awareness among individuals. More details can be found in:

Tiffany C.K. Kwok, **Michael Xuelin Huang**, Wai Cheong Tam and Grace Ngai. "Emotar: Communicating Feelings through Video Sharing". Proceedings of the 20th International Conference on Intelligent User Interfaces *IUI '15*. ACM Press, 374-378.

Figure 7-2. Screenshot of Emotar interface.

7.4.3 Multimodal attention detection

This work attempts to detect the user's attention during normal human-computer interactions through a noninvasive multimodal solution, which allows participants to work naturally without interference. The solution uses webcam, keyboard, and mouse. These modalities could reasonably be expected of any computing environment and does not rely on expensive and tailor-made equipment. For more information see:

Sun, Hugo Jiawei, **Huang, Michael Xuelin**, Ngai, Grace, Chan, Stephen Chi Fai. 2014. Nonintrusive Multimodal Attention Detection**.** *The 7th International Conference on Advances in Computer-Human Interactions ACHI 2014*.

7.4.4 Physiological mouse

We propose to make use of human physiological signals in determining human affects in a non-intrusive manner. This is achieved via the physiological mouse, as the first step towards affective computing. We augment the mouse with a small optical component for capturing user photoplethysmogram (c) signal. With the PPG signal, we are able to compute and derive human physiological signals. More details can be found in:

Y. Fu, H.V. Leong, G. Ngai, **Huang, Michael Xuelin**, S.C.F. Chan. "Physiological mouse: Towards an emotion-aware mouse". In Universal Access in the Information Society: An International Journal, Springer

Fu, Yujun, Leong, Hong Va, Ngai, Grace, **Huang, Michael Xuelin**, Chan Stephen. 2014. "Physiological Mouse: Towards an Emotion-Aware Mouse". *1st IEEE International Workshop on User Centered Design and Adaptive Systems UCDAS 2014.*

7.4.5 Mobile DJ

This work presents Mobile DJ, a tangible, mobile platform for active music listening, designed to augment internet-based social interaction with the element of active music listening. A tangible interface facilitates users to manipulate musical effects. Multiple users with the internet connect can collaborate and interact through their music. User tests indicate that the device is successful at allowing user immersion into the active listening experience, and that users enjoy the added sensory input as well as the novel way of interacting with the music and each other. More details can be found in:

Lo, Kenneth W.K., Lau, Chi Kin, **Huang, Michael Xuelin**, Tang, Wai Wa, Ngai, Grace, Chan, Stephen C.F. "Mobile DJ: a Tangible, Mobile Platform for Active and Collaborative Music Listening". *NIME '13*. 2013.

References

[1] Aigrain, J., Dubuisson, S., Detyniecki, M. and Chetouani, M. 2015. Person-specific behavioural features for automatic stress detection. *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition* (2015).

[2] Alnajar, F., Gevers, T., Valenti, R. and Ghebreab, S. 2013. Calibration-Free Gaze Estimation Using Human Gaze Patterns. *2013 IEEE International Conference on Computer Vision* (Dec. 2013), 137–144.

[3] Altman, N.S. 1992. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*. 46, (1992), 175–185.

[4] Andreassi, J.L. 2006. *Psychophysiology: Human Behavior and Physiological Response*. Psychology Press.

[5] Andreu-Perez, J., Solnais, C. and Sriskandarajah, K. 2016. EALab (Eye Activity Lab): a MATLAB Toolbox for Variable Extraction, Multivariate Analysis and Classification of Eye-Movement Data. *Neuroinformatics*. 14, (2016), 51–67.

[6] Ashraf, A.B., Lucey, S., Cohn, J.F., Chen, T., Ambadar, Z., Prkachin, K.M. and Solomon, P.E. 2009. The painful face – Pain expression recognition using active appearance models. *Image and Vision Computing*. 27, (2009), 1788–1796.

[7] Bartlett, M.S., Littlewort, G.C., Frank, M.G., Lainscsek, C., Fasel, I.R. and Movellan, J.R. 2006. Automatic Recognition of Facial Actions in Spontaneous Expressions. *Journal of Multimedia*. 1, 6 (Sep. 2006).

[8] Breiman, L. 2001. Random forests. *Machine Learning*. 45, (2001), 5–32.

[9] Calvo, R.A. and D'Mello, S. 2010. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*. 1, 1 (2010), 18–37.

[10] Çeliktutan, O. and Gunes, H. 2014. Continuous prediction of perceived traits and social dimensions in space and time. *2014 IEEE International Conference on Image Processing, ICIP 2014* (2014), 4196–4200.

[11] Le Cessie, S. and van Houwelingen, J.C. 1992. Ridge Estimators in Logistic Regression. *Applied Statistics*. 41, 1 (1992), 191–201.

[12] Chang, C.-C. and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2, 3 (Apr. 2011), 1–27.

[13] Chen, J., Liu, X., Tu, P. and Aragones, A. 2013. Learning person-specific models for facial expression and action unit recognition. *Pattern Recognition Letters*. 34, 15 (Nov. 2013), 1964–1970.

[14] Chen, M.C., Anderson, J.R. and Sohn, M.H. 2001. What can a mouse cursor tell us more? *CHI '01 extended abstracts on Human factors in computing systems - CHI '01* (New York, New York, USA, 2001), 281.

[15] Chen, Y., Bi, J. and Wang, J.Z. 2006. MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 28, (2006), 1931–1947.

[16] Cheplygina, V., Tax, D.M.J. and Loog, M. 2014. Dissimilarity-based Ensembles for Multiple Instance Learning. *IEEE Transactions on Neural Networks and Learning Systems*. (2014), 1–12.

[17] Cheung, Y. ming and Peng, Q. 2015. Eye Gaze Tracking With a Web Camera in a Desktop Environment. *IEEE Transactions on Human-Machine Systems*. (2015).

[18] Chu, W.-S., De La Torre, F. and Cohn, J.F. 2013. Selective Transfer Machine for Personalized Facial Action Unit Detection. *2013 IEEE Conference on Computer Vision and Pattern Recognition* (Jun. 2013), 3515–3522.

[19] Cohen, I., Sebe, N., Garg, A., Chen, L.S. and Huang, T.S. 2003. Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and Image Understanding*. 91, 1-2 (Jul. 2003), 160–187.

[20] Cohn, J.F. 2006. Foundations of human computing: facial expression and emotion. *Artifical Intelligence for Human Computing*. 2006, (2006), 1–16.

[21] Cootes, T.F., Edwards, G.J. and Taylor, C.J. 2001. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 23, 6 (Jun. 2001), 681–685.

[22] Cootes, T.F. and Taylor, C.J. 1992. Active Shape Models - “Smart Snakes.” *BMVC92* (1992), 266–275.

[23] Corneanu, C.A., Oliu, M., Cohn, J.F. and Escalera, S. 2016. Survey on RGB, 3D, Thermal, and Multimodal Approaches for Facial Expression Recognition:

History, Trends, and Affect-related Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 8828, (2016), 1–1.

[24] Dai, W., Yang, Q., Xue, G.-R. and Yu, Y. 2007. Boosting for Transfer Learning. *Proceedings of the 24th International Conference on Machine learning - ICML '07*. (2007), 193–200.

[25] Dhall, A., Goecke, R., Joshi, J., Sikka, K. and Gedeon, T. 2014. Emotion Recognition In The Wild Challenge 2014. *Proceedings of the 16th International Conference on Multimodal Interaction - ICMI '14* (2014), 461–466.

[26] Dhall, A., Goecke, R., Lucey, S. and Gedeon, T. 2012. Collecting large, richly annotated facial-expression databases from movies. *IEEE Multimedia*. 19, (2012), 34–41.

[27] Ekman, P. and Friesen, W. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press.

[28] Fares, R., Fang, S. and Komogortsev, O. 2013. Can we beat the mouse with MAGIC? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (New York, New York, USA, 2013), 1387.

[29] Fontaine, J.R.J., Scherer, K.R., Roesch, E.B. and Ellsworth, P.C. 2007. The World of Emotions Is Not Two-Dimensional. *Psychological Science*. 18, (2007), 1050–1057.

[30] Foulds, J. and Frank, E. 2010. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*. 25, 01 (Mar. 2010), 1.

[31] Freund, Y. and Schapire, R. 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*. 55, (1995), 119–139.

[32] Fu, Y., Leong, H.V., Ngai, G., Huang, M.X. and Chan, S.C.F. 2014. Physiological Mouse: Towards an Emotion-Aware Mouse. *2014 IEEE 38th International Computer Software and Applications Conference Workshops* (2014), 258–263.

[33] Fu, Z., Robles-Kelly, A. and Zhou, J. 2011. MILIS: Multiple instance learning with instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 33, (2011), 958–977.

[34] Gingerich, M. and Conati, C. 2015. Constructing Models of User and Task Characteristics from Eye Gaze Data for User-Adaptive Information

Highlighting. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015), 1728–1734.

[35] Gross, J.J. and Levenson, R.W. 1995. Emotion elicitation using films. *Cognition & Emotion*.

[36] Gross, R., Matthews, I., Cohn, J., Kanade, T. and Baker, S. 2008. Multi-PIE. *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition* (Sep. 2008), 1–8.

[37] Guo, Q. and Agichtein, E. 2010. Towards predicting web searcher gaze position from mouse movements. *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems - CHI EA '10* (New York, New York, USA, 2010), 3601.

[38] Hamerly, G. and Elkan, C. 2003. Learning the k in k-means. *In Advances in Neural Information Processing Systems (NIPS)*.

[39] Hansen, D.W. and Ji, Q. 2010. In the eye of the beholder: a survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence*. 32, 3 (Mar. 2010), 478–500.

[40] Hernandez, J., Paredes, P., Roseway, A. and Czerwinski, M. 2014. Under pressure: Sensing Stress of Computer Users. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (2014), 51–60.

[41] Hoque, M.E., McDuff, D.J. and Picard, R.W. 2012. Exploring Temporal Patterns in Classifying Frustrated and Delighted Smiles. *IEEE Transactions on Affective Computing*. 3, 3 (Jul. 2012), 323–334.

[42] Hornof, A.J. and Halverson, T. 2002. Cleaning up systematic error in eye-tracking data by using required fixation locations. *Behavior research methods, instruments, & computers : a journal of the Psychonomic Society, Inc*. 34, (2002), 592–604.

[43] Huang, J., White, R. and Buscher, G. 2012. User see, user point: gaze and cursor alignment in web search. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12* (New York, New York, USA, 2012), 1341.

[44] Huang, J., White, R.W. and Dumais, S. 2011. No clicks, no problem: using cursor movements to understand and improve search. *Proceedings of the 2011*

annual conference on Human factors in computing systems - CHI '11 (New York, New York, USA, 2011), 1225.

[45] Huang, M.X., Kwok, T.C.K., Ngai, G., Leong, H.V. and Chan, C.F.S. 2014. Building a Self-Learning Eye Gaze Model from User Interaction Data. *ACM Multimedia* (2014).

[46] Huang, M.X., Ngai, G., Hua, K.A. and Chan, S.C.F. Identifying User-Specific Facial Affects from Spontaneous Expressions with Minimal Annotation. *To appear in Transactions on Affective Computing*.

[47] Huang, X., Kortelainen, J., Zhao, G., Li, X., Moilanen, A., Sepplanen, T. and Pietiklainen, M. 2016. Multi-modal Emotion Analysis from Facial Expressions and Electroencephalogram. *Computer Vision and Image Understanding*. 147, (2016), 114–124.

[48] Jacob, R.J.K. 1990. What you look at is what you get: eye movement-based interaction techniques. *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90* (New York, New York, USA, 1990), 11–18.

[49] Jaques, N., Conati, C., Harley, J.M. and Azevedo, R. 2014. Predicting affect from gaze data during interaction with an intelligent tutoring system. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2014), 29–38.

[50] Jia, J., Wu, Z., Zhang, S., Meng, H.M. and Cai, L. 2014. Head and facial gestures synthesis using PAD model for an expressive talking avatar. *Multimedia Tools and Applications*. 73, (2014), 439–461.

[51] Jianbo Shi and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 22, (2000), 888–905.

[52] Jones, K.S. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*. 28, 1 (1972), 11–21.

[53] Judd, T., Ehinger, K., Durand, F. and Torralba, A. 2009. Learning to predict where humans look. *2009 IEEE 12th International Conference on Computer Vision* (Sep. 2009), 2106–2113.

[54] El Kaliouby, R. and Robinson, P. 2005. Generalization of a Vision-Based Computational Model of Mind-Reading. *ACII'05 Proceedings of the First*

international conference on Affective Computing and Intelligent Interaction (2005), 582–589.

[55] El Kaliouby, R. and Robinson, P. 2004. Real-Time Inference of Complex Mental States from Facial Expressions and Head Gestures. *2004 Conference on Computer Vision and Pattern Recognition Workshop* (2004), 154–154.

[56] Keerthi, S.S., Shevade, S.K., Bhattacharyya, C. and Murthy, K.R.K. 2001. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation*. 13, (2001), 637–649.

[57] Koelstra, S., Mühl, C., Soleymani, M., Lee, J.S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A. and Patras, I. 2012. DEAP: A database for emotion analysis; Using physiological signals. *IEEE Transactions on Affective Computing*. 3, (2012), 18–31.

[58] Koldijk, S., Sappelli, M., Verberne, S., Neerincx, M.A. and Kraaij, W. 2014. The SWELL Knowledge Work Dataset for Stress and User Modeling Research. *Proceedings of the 16th International Conference on Multimodal Interaction* (2014), 291–298.

[59] De la Torre, F., Campoy, J., Ambadar, Z. and Cohn, J.F. 2007. Temporal Segmentation of Facial Behavior. *2007 IEEE 11th International Conference on Computer Vision*. (2007), 1–8.

[60] De la Torre, F., Simon, T., Ambadar, Z. and Cohn, J.F. 2011. FAST-FACS : A Computer-Assisted System to Increase Speed and Reliability of Manual FACS Coding. *Affective Computing and Intelligent Interaction, 2011* (2011), 1–10.

[61] Leistner, C., Saffari, A. and Bischof, H. 2010. MIForests: Multiple-instance learning with randomized trees. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2010), 29–42.

[62] Li, Y., Mavadati, S.M., Mahoor, M.H. and Ji, Q. 2013. A unified probabilistic framework for measuring the intensity of spontaneous facial action units. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2013* (2013).

[63] Liang, G., Cao, J., Liu, X. and Han, X. 2014. Cushionware. *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA '14* (2014), 591–594.

[64] Liebling, D.J. and Dumais, S.T. 2014. Gaze and mouse coordination in everyday work. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct* (2014), 1141–1150.

[65] Littlewort, G., Bartlett, M.S., Fasel, I., Susskind, J. and Movellan, J. 2004. Dynamics of Facial Expression Extracted Automatically from Video. *Proc. of Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04* (2004), 80.

[66] Littlewort, G.C., Bartlett, M.S. and Lee, K. 2007. Faces of pain: automated measurement of spontaneousallfacial expressions of genuine and posed pain. *Proceedings of the ninth international conference on Multimodal interfaces - ICMI '07* (New York, New York, USA, 2007), 15.

[67] Lu, F., Okabe, T., Sugano, Y. and Sato, Y. 2011. A Head Pose-free Approach for Appearance-based Gaze Estimation. *Procedings of the British Machine Vision Conference 2011* (2011), 126.1–126.11.

[68] Lu, F., Okabe, T., Sugano, Y. and Sato, Y. 2014. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing*. 32, 3 (Mar. 2014), 169–179.

[69] Lu, F., Sugano, Y., Okabe, T. and Sato, Y. 2015. Gaze Estimation From Eye Appearance: A Head Pose-Free Method via Eye Image Synthesis. *IEEE Transactions on Image Processing*. 24, (2015), 3680–3693.

[70] Lu, F., Sugano, Y., Okabe, T. and Sato, Y. 2014. Inferring Human Gaze from Appearance via Adaptive Linear Regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (2014), 1–1.

[71] Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z. and Matthews, I. 2010. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops* (Jun. 2010), 94–101.

[72] Lucey, P., Cohn, J.F., Prkachin, K.M., Solomon, P.E. and Matthews, I. 2011. Painful data: The UNBC-McMaster shoulder pain expression archive database. *Face and Gesture 2011* (Mar. 2011), 57–64.

[73] Lundberg, U., Kadefors, R., Melin, B., Palmerud, G., Hassmen, P., Engstrom, M. and Dohns, I.E. 1994. Psychophysiological stress and EMG activity of the

trapezius muscle. *International journal of behavioral medicine*. 1, (1994), 354–370.

[74] Lyu, Y., Luo, X., Zhou, J., Yu, C., Miao, C., Wang, T., Shi, Y. and Kameyama, K. 2015. Measuring Photoplethysmogram-Based Stress-Induced Vascular Response Index to Assess Cognitive Load and Stress. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15* (2015), 857–866.

[75] MacQueen, J.B. 1967. Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability* (1967), 281–297.

[76] Mavadati, S.M., Mahoor, M.H., Bartlett, K., Trinh, P. and Cohn, J.F. 2013. DISFA: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*. 4, (2013), 151–160.

[77] McDuff, D., El Kaliouby, R., Kassam, K. and Picard, R. 2010. Affect valence inference from facial action unit spectrograms. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. (Jun. 2010), 17–24.

[78] Michel, P. and El Kaliouby, R. 2003. Real time facial expression recognition in video using support vector machines. *Proceedings of the 5th international conference on Multimodal interfaces - ICMI '03* (New York, New York, USA, 2003), 258.

[79] Mou, W., Celiktutan, O. and Gunes, H. 2015. Group-level arousal and valence recognition in static images: Face, body and context. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. (2015), 1–6.

[80] Nicolaou, M.A., Member, S. and Gunes, H. 2011. Continuous Prediction of Spontaneous Affect from Multiple Cues and Modalities in Valence-Arousal Space. 2, 2 (2011), 92–105.

[81] Oertel, C. and Salvi, G. 2013. A gaze-based method for relating group involvement to individual engagement in multimodal multiparty dialogue. *Proceedings of the 15th ACM on International conference on multimodal interaction - ICMI '13* (2013), 99–106.

[82] Pan, S.J. and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*.

[83] Paul Viola, M.J. 2001. Robust Real-time Object Detection. *International Journal of Computer Vision*. (2001).

[84] Pentel, A. 2015. Patterns of Confusion: Using Mouse Logs to Predict User's Emotional State. *Conference: 5th International Workshop on Personalization Approaches in Learning Environments (PALE 2015) in conjunction with 23rd Conference on User Modelling, Adaptation and Personalization (UMAP 2015)* (2015).

[85] Picard, R.W. 1997. *Affective Computing*. MIT Press.

[86] Platt, J.C. 1999. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. *Advances in kernel methods*. 185 – 208.

[87] Prkachin, K.M. and Solomon, P.E. 2008. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*. 139, (2008), 267–274.

[88] Reichle, E.D., Reineberg, A.E. and Schooler, J.W. 2010. Eye Movements During Mindless Reading. *Psychological Science*. 21, (2010), 1300–1310.

[89] Rodden, K., Fu, X., Aula, A. and Spiro, I. 2008. Eye-mouse coordination patterns on web search results pages. *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08* (New York, New York, USA, 2008), 2997.

[90] Rodrigue, M., Son, J., Giesbrecht, B., Turk, M. and Höllerer, T. 2015. Spatio-Temporal Detection of Divided Attention in Reading Applications Using EEG and Eye Tracking. *Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI '15* (2015), 121–125.

[91] Rosenberg, E.L. and Ekman, P. 1994. Coherence between expressive and experiential systems in emotion. *Cognition & Emotion*. 8, 3 (May 1994), 201–229.

[92] Ruiz, A., Weijer, J. Van de and Binefa, X. 2014. Regularized Multi-Concept MIL for weakly-supervised facial behavior categorization. *Proceedings of the British Machine Vision Conference* (2014).

[93] Salvucci, D.D., Salvucci, D.D., Goldberg, J.H. and Goldberg, J.H. 2000. Identifying Fixations and Saccades in Eye-Tracking Protocols. *Proceedings of the Eye Tracking Research and Applications Symposium*. (2000), 71–78.

[94] Sangineto, E., Zen, G., Ricci, E. and Sebe, N. 2014. We are not All Equal: Personalizing Models for Facial Expression Analysis with Transductive

Parameter Transfer. *Proceedings of the ACM International Conference on Multimedia - MM '14* (2014), 357–366.

[95] Saragih, J.M., Lucey, S. and Cohn, J.F. 2010. Deformable Model Fitting by Regularized Landmark Mean-Shift. *International Journal of Computer Vision*. 91, 2 (Sep. 2010), 200–215.

[96] Saragih, J.M., Lucey, S. and Cohn, J.F. 2009. Face alignment through subspace constrained mean-shifts. *2009 IEEE 12th International Conference on Computer Vision* (Sep. 2009), 1034–1041.

[97] Sariyanidi, E., Gunes, H. and Cavallaro, A. 2014. Automatic analysis of facial affect: A survey of registration, representation and recognition. *TPAMI IEEE Transactions on Pattern Analysis and Machine Intelligence*. 8828, (2014), 1–22.

[98] Sikka, K., Dhall, A. and Bartlett, M. 2013. Weakly supervised pain localization using multiple instance learning. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)* (Apr. 2013), 1–8.

[99] Sikka, K., Dhall, A. and Bartlett, M.S. 2014. Classification and weakly supervised pain localization using multiple segment representation. *Image and Vision Computing*. 32, 10 (2014), 659–670.

[100] Sikka, K., Wu, T., Susskind, J. and Bartlett, M. 2012. Exploring bag of words architectures in the facial expression domain. *ECCV 2012* (2012), 250–259.

[101] Soleymani, M., Lichtenauer, J., Pun, T. and Pantic, M. 2012. A multimodal database for affect recognition and implicit tagging. *IEEE Transactions on Affective Computing*. 3, (2012), 42–55.

[102] Sugano, Y., Matsushita, Y. and Sato, Y. 2013. Appearance-based gaze estimation using visual saliency. *IEEE transactions on pattern analysis and machine intelligence*. 35, 2 (Feb. 2013), 329–41.

[103] Sugano, Y., Matsushita, Y., Sato, Y. and Koike, H. 2008. An Incremental Learning Method for Unconstrained Gaze Estimation. *10th European Conference on Computer Vision, ECCV' 2008* (2008), 656–667.

[104] Sun, D., Paredes, P. and Canny, J. 2014. MouStress: Detecting Stress from Mouse Motion. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (2014), 61–70.

[105] Valenti, R., Sebe, N. and Gevers, T. 2012. Combining head pose and eye location information for gaze estimation. *IEEE Transactions on Image Processing*. 21, (2012), 802–815.

[106] Valstar, M., Schuller, B., Smith, K., Almaev, T., Eyben, F., Krajewski, J., Cowie, R. and Pantic, M. 2014. AVEC 2014. *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge - AVEC '14* (2014), 3–10.

[107] Valstar, M.F., Mehu, M., Jiang, B., Pantic, M. and Scherer, K. 2012. Meta-Analysis of the First Facial Expression Recognition Challenge. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*. (Jun. 2012).

[108] Vinciarelli, A. and Mohammadi, G. 2014. A survey of personality computing. *IEEE Transactions on Affective Computing*.

[109] Viola, P. and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 1, (2001), I–511–I–518.

[110] Viola, P., Platt, J.C. and Zhang, C. 2006. Multiple Instance Boosting for Object Detection. *Advances in neural information processing systems* (2006), 1417.

[111] Vizer, L.M., Zhou, L. and Sears, A. 2009. Automated stress detection using keystroke and linguistic features: An exploratory study. *International Journal of Human Computer Studies*. 67, (2009), 870–886.

[112] Wahlstrom, J., Hagberg, M., Johnson, P.W., Svensson, J. and Rempel, D. 2002. Influence of time pressure and verbal provocation on physiological and psychological reactions during work with a computer mouse. *European Journal of Applied Physiology*. 87, (2002), 257–263.

[113] Williams, O., Blake, A. and Cipolla, R. Sparse and Semi-supervised Visual Mapping with the S^3GP. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)* 230–237.

[114] Wood, E., Baltrušaitis, T., Morency, L.-P., Robinson, P. and Bulling, A. 2016. Learning an appearance-based gaze estimator from one million synthesised images. *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16* (2016), 131–138.

[115] Wood, E., Baltrusaitis, T., Zhang, X., Sugano, Y., Robinson, P. and Bulling, A. 2015. Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. *IEEE International Conference on Computer Vision (ICCV)* (2015), 3756–3764.

[116] Wood, E. and Bulling, A. 2014. EyeTab: Model-based gaze estimation on unmodified tablet computers. *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14* (New York, New York, USA, 2014), 207–210.

[117] Xiao, Y., Liu, B., Hao, Z. and Cao, L. 2014. A similarity-based classification framework for multiple-instance learning. *IEEE transactions on cybernetics*. 44, 4 (Apr. 2014), 500–15.

[118] Xiong, X. and De la Torre, F. 2013. Supervised Descent Method and Its Applications to Face Alignment. *2013 IEEE Conference on Computer Vision and Pattern Recognition* (Jun. 2013), 532–539.

[119] Xu, L. and Mordohai, P. 2010. Automatic Facial Expression Recognition using Bags of Motion Words. *Procedings of the British Machine Vision Conference 2010* (2010), 13.1–13.13.

[120] Zen, G., Sangineto, E. and Ricci, E. 2014. Unsupervised Domain Adaptation for Personalized Facial Emotion Recognition. *ICMI* (2014).

[121] Zeng, Z., Pantic, M., Roisman, G.I. and Huang, T.S. 2009. A survey of affect recognition methods: audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*. 31, 1 (Jan. 2009), 39–58.

[122] Zhang, L., Tong, Y. and Ji, Q. 2008. Active Image Labeling and Its Application to Facial Action Labeling. *ECCV '08 Proceedings of the 10th European Conference on Computer Vision* (2008).

[123] Zhang, X., Sugano, Y., Fritz, M. and Bulling, A. 2015. Appearance-Based Gaze Estimation in the Wild. *28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)* (2015), 9.

[124] Zhang, X., Yin, L., Cohn, J.F., Canavan, S., Reale, M., Horowitz, A., Liu, P. and Girard, J.M. 2014. BP4D-Spontaneous: a high-resolution spontaneous 3D dynamic facial expression database. *Image and Vision Computing*.

[125] Zhang, Y. and Hornof, A.J. 2014. Easy post-hoc spatial recalibration of eye tracking data. *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14* (2014), 95–98.

[126] Zhang, Y. and Hornof, A.J. 2011. Mode-of-disparities error correction of eye-tracking data. *Behavior research methods*. 43, (2011), 834–842.

[127] Zhou, F., De la Torre, F. and Cohn, J.F. 2010. Unsupervised discovery of facial events. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. (Jun. 2010), 2574–2581.

[128] Zhu, J. and Yang, J. 2002. Subpixel eye gaze tracking. *Fifth IEEE International Conference on Automatic Face Gesture Recognition* (2002), 131–136.

[129] Zhu, Y., De la Torre, F., Zhang, Y.-J. and Cohn, J.F. 2011. Dynamic Cascades with Bidirectional Bootstrapping for Action Unit Detection in Spontaneous Facial Behavior. *IEEE Transactions on Affective Computing*. 2, 2 (Apr. 2011), 79–91.