Pattern Recognition 60 (2016) 890–900

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Facial action unit recognition under incomplete data based on multi-label learning with missing labels

Yongqiang Li [a,c], Baoyuan Wu [b,*], Bernard Ghanem [b], Yongping Zhao [a], Hongxun Yao [c], Qiang Ji [d]

[a] School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin 15001, China
[b] The Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Kingdom of Saudi Arabia
[c] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 15001, China
[d] Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

ARTICLE INFO

Article history:
Received 6 April 2016
Received in revised form
10 June 2016
Accepted 4 July 2016
Available online 7 July 2016

Keywords:
Face action unit recognition
Incomplete data
Multi-label learning

ABSTRACT

Facial action unit (AU) recognition has been applied in a wild range of fields, and has attracted great attention in the past two decades. Most existing works on AU recognition assumed that the complete label assignment for each training image is available, which is often not the case in practice. Labeling AU is expensive and time consuming process. Moreover, due to the AU ambiguity and subjective difference, some AUs are difficult to label reliably and confidently. Many AU recognition works try to train the classifier for each AU independently, which is of high computation cost and ignores the dependency among different AUs. In this work, we formulate AU recognition under incomplete data as a multi-label learning with missing labels (MLML) problem. Most existing MLML methods usually employ the same features for all classes. However, we find this setting is unreasonable in AU recognition, as the occurrence of different AUs produce changes of skin surface displacement or face appearance in different face regions. If using the shared features for all AUs, much noise will be involved due to the occurrence of other AUs. Consequently, the changes of the specific AUs cannot be clearly highlighted, leading to the performance degradation. Instead, we propose to extract the most discriminative features for each AU individually, which are learned by the supervised learning method. The learned features are further embedded into the instance-level label smoothness term of our model, which also includes the label consistency and the class-level label smoothness. Both a global solution using st-cut and an approximated solution using conjugate gradient (CG) descent are provided. Experiments on both posed and spontaneous facial expression databases demonstrate the superiority of the proposed method in comparison with several state-of-the-art works.

1. Introduction

Facial expression is one of the most natural nonverbal communication media that individuals use to regulate interactions with each other. Expressions can express the emotions, clarify and emphasize what is being said, and signal comprehension, disagreement and intentions [7]. Machine understanding of facial expressions will provide powerful information to describe the emotional states and psychological patterns of individuals. Due to the huge potential in many applications, including social robotics, affective online tutoring environment, intelligent Human–Computer interaction (HCI), etc., automatic facial expression recognition has recently gained great attention and become a hot topic [2,3].

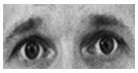
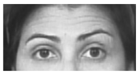
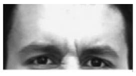
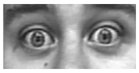
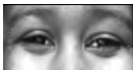
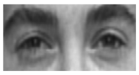
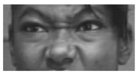
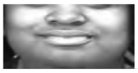
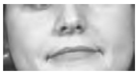
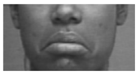
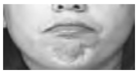
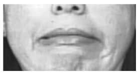
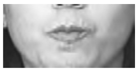
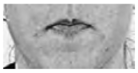
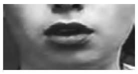
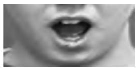
One of the most widely studied expression descriptors is the six basic expressions named anger, fear, disgust, happiness, sadness and surprise, which are universal and unrelated with race and culture [4]. However, these six basic expressions only represent a small set of human facial expressions. In fact, human emotion is composed of thousands of expressions, though most of them differ in subtle changes of a few facial features. Facial Action Coding System (FACS) developed by Ekman [5] has been demonstrated as a powerful means for representing and characterizing a large number of facial expressions through the combination of only a small set of action units (AUs). According to FACS, each AU is related to the contraction of a specific set of facial muscles. FACS defines 32 AUs, i.e., 9 AUs in the upper face, 18 in the lower face and 5 AUs that cannot be partitioned as belonging to either the upper or the lower face [7]. The readers are referred to **[5]** for detailed definition and explanation of all AUs. We list the 16 AUs we recognize in this work in Table 1.[1] The aim of AU recognition is

* Corresponding author.
E-mail address: wubaoyuan1987@gmail.com (B. Wu).

[1] Readers are referred to: http://www.cs.cmu.edu/face/facs.htm, where the pictures, facial muscles and descriptions of all AUs are listed.

http://dx.doi.org/10.1016/j.patcog.2016.07.009
0031-3203/© 2016 Elsevier Ltd. All rights reserved.

Table 1
A list of several frequent AUs, their interpretations, corresponding face regions and facial muscles. (adapted from [5]).

AUs	Picture	Interpretation	Facial Muscles	AUs	Picture	Interpretation	Facial Muscles
AU1		Inner Brow Raiser	Frontalis, pars medialis	AU2		Outer Brow Raiser	Frontalis, pars lateralis
AU4		Brow Lowerer	Corrugator supercilii, Depressor supercilii	AU5		Upper Lid Raiser	Levator palpebrae superioris
AU6		Cheek Raiser	Orbicularis oculi, pars orbitalis	AU7		Lid Tightener	Orbicularis oculi, pars palpebralis
AU9		Nose Wrinkler	Levator labii superioris alaquae nasi	AU12		Lip Corner Puller	Zygomaticus Major
AU14		Dimpler	Buccinator	AU15		Lip Corner Depressor	Depressor anguli oris
AU17		Chin Raiser	Mentalis	AU20		Lip Stretcher	Risorius platysma
AU23		Lip Tightener	Orbicularis oris	AU24		Lip Pressor	Orbicularis oris
AU25		Lip Part	Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris	AU27		Mouth Stretch	Pterygoids, Digastric

to recognize all present AUs from expressional images, and then to describe all possible facial expressions. More formally, an expressional image is denoted as an instance x, and m candidate AUs are represented as $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$. The label vector of x is denoted as $z \in \{-1, +1\}^m$, where a positive value indicates the presence of the corresponding class (AU), while a negative value means absence. Consequently, the AU recognition can be seen as the prediction of the complete label vector z of x, which is naturally formulated as a multi-label learning problem.

The majority of previous multi-label learning methods assume that each training instance is associated with a complete label assignment. However, in AU recognition, it is often difficult to obtain a complete label assignment for each training sample. For example, AU is typically manually labeled by trained human experts, which is expensive and time consuming. Furthermore, because of the ambiguity nature of AUs as well as the subjective difference, some AUs are difficult to label reliably and confidently. Hence a more realistic scenario is that we have to learn the model from incomplete data, i.e., a part of labels for some training instances are missing. To explicitly accommodate the missing labels, we utilize the definition of an incomplete label vector $y \in \{-1, 0, +1\}^m$ where a 0 indicates the missing label.

Wu et al. [8,9] proposed a multi-label learning with missing labels strategy for AU recognition, and achieved improved performance compared to several state-of-the-art methods. They utilize both the instance-level and class-level smoothness to build a unified graph, such that the label information can be propagated from the provided labels to missing labels. However, similar as most multi-label learning methods [12,13,43–44], studies [8,9] compute a shared instance-level similarity for all AU classes between two instances, based on the whole features of images. According to FACS, different AUs are caused by different sets of facial muscles, and hence produce changes of skin surface displacement or face appearance in different face regions. For example, as shown in Table 1, the contraction of muscle group *Occipito Frontalis* will produce AU1 while the contraction of muscle group *Mentalis* will cause AU15. These two AUs cause feature changes in different regions. Hence, features selected for AU1 are not discriminative for AU15, and vice verse (as demonstrated in Fig. 1). Thus, we believe that the shared instance-level similarity for all AU classes is unable to distinguish the subtle changes of different AUs.

In this work, we propose to firstly learn the discriminative features for each AU class by supervised learning. As a result, the feature noise from the occurrence of other AUs could be filtered. The class-specific instance-level similarity among instances is then computed based on the learned features, which is further incorporated into the MLML model. The proposed model investigates both the discriminative information from features and the label dependency information from labels in a principled manner. We also provide two efficient solutions, including the exact solution based on ST-CUT method [56] and the approximate solution based on conjugate gradient method [55]. Sufficient evaluations on both posed and spontaneous expression databases demonstrate the superiority of the proposed method compared to state-of-the-art.

2. Related works

Because of great potential application value in a wide range of different fields, there has been extensive research on facial expression analysis over the past decades. Typical strategy for facial actions recognition includes two phases: image feature extraction and machine analysis of facial actions. For image feature extraction, some works try to extract features based on the location of facial salient points [26,27], and the shapes of the facial components [28,29,1], which are usually referred as geometric features. Face surface and skin texture changes such as wrinkles, bulges, and furrows are employed as appearance features. Such kind of features includes Gabor feature [30], Haar feature [31,32], LBP feature [33,34], etc. Detailed survey on AU recognition is referred to [18].

Machine analysis methods can be grouped into separated-AU recognition methods and joint-AU recognition methods.

Separated-AU recognition methods usually try to learn discriminative classifier for each AU individually [31,35,30,26,27], which ignore the correlation dependency among AUs. In contrast,

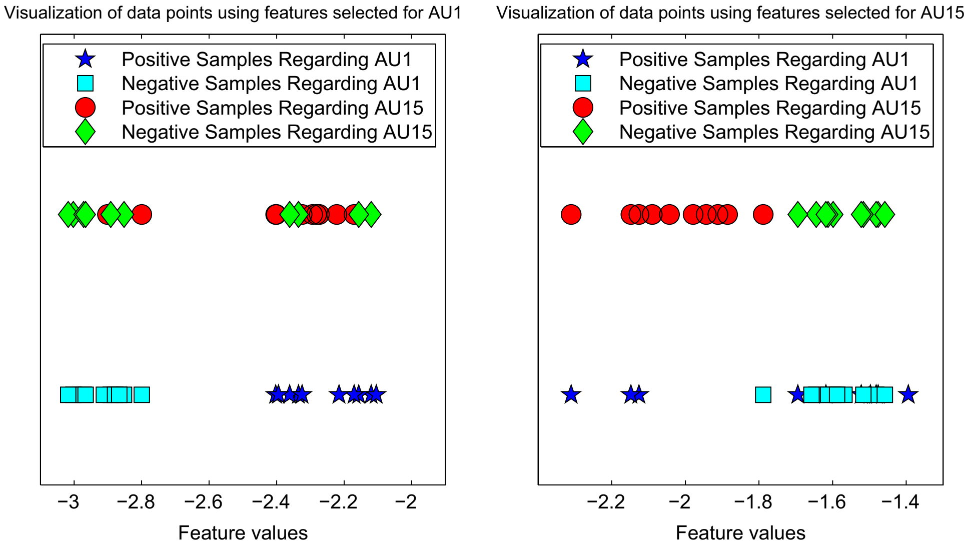


Fig. 1. An demonstration of features selected for one AU but used to discriminate another AU.

joint-AU recognition methods model the semantics among AUs and recognize several AUs jointly. Dynamic Bayesian Networks (DBN) [14,37] are firstly and extensively investigated for this purpose. A downside of DBN based framework [14,37,38] is modeling the feature discriminative information and label correlation information independently, which could result in inconsistent dependences across inputs/outputs. Recent works [49–52] all address jointly AU recognition and formulate discriminative learning and AU dependency learning in a single model. Work [49] employs a restricted Boltzman machine (RBM) where latent variables account for the dependencies among AUs and are directly linked to the image features. Moreover, [50,51] combine multi-task learning with MKL to jointly learn different AU classifiers. Recent work [52] first projects different kinds of features onto a shared manifold, and learns logistic functions based the manifold for multi AU recognition simultaneously. The common drawback of works [49–52] is discriminating all target AUs based on same features. However, different AUs produce feature changes in different face regions, and hence using all features could involve much noise from the occurrence of other AUs.

Besides, current machine analysis methods for AU recognition usually assume that there are complete label assignments for each training sample. However, a more realistic scenario is that each image is only provided with a partial label assignment, i.e., incomplete data. The classical approaches for learning under incomplete data include Expectation Maximization (EM) algorithm [39] and Gibbs sampling [40]. However, when data are missing completely at random, those methods could fail, where the learned parameters may be quite different from the true parameters. Liao and Ji [41] learn Bayesian Networks under incomplete data by making use of qualitative constraints, which is then applied to AU recognition. However, study [41] needs domain experts to preset the corresponding qualitative constraints among AUs.

In this work, we formulate AU recognition under incomplete data as a multi-label learning with missing labels problem. In the literature of multi-label learning, there have been some investigations on learning with missing labels. A commonly used strategy is to treat the missing labels as negative labels [13,42–44]. For instance, study [13] addresses the special case when training samples are either fully labeled or completely unlabeled. Study [44] focuses on the case that training samples only have a partial set of positive labels available while the rest of the labels are unassigned. Both works [13] and [44] set the missing labels to zero by default, which coincides with the numerical value assigned to negative labels. The obvious drawback of those methods is that the label bias is involved. Another strategy is to treat filling in missing labels as a matrix completion (MC) problem such as studies [45–47]. The basic idea is to concatenate the label matrix and feature matrix into a unified matrix, and then the standard matrix completion techniques can be applied to fill in the missing labels. The basic assumption of those works is based on the low rank assumption, which is widely used but may not hold in practical multi-label problem.

Besides the limitations in dealing with missing labels, another significant drawback of current multi-label learning methods is using the same features to discriminate all the classes, such as studies [8,9]. However, for AU recognition, since the occurrence of different AUs produces changes of skin surface displacement or face appearance in different face regions, using the same features (all features) for all AUs will involve much noise from the occurrence of other AUs, and hence limit the model performance. A more reasonable strategy is to individually extract features for each AU from the locations where the AU is relatively fixed. In this work, we model the feature similarity for each AU only based on the most related features.

In summary, the contributions of this work include:

1) We formulate AU recognition under incomplete data as a MLML problem, which deals with the missing labels in a principled manner.
2) Different from previous MLML works, which usually use same features for all classes, we discriminate each AU based on the most related features and hence significantly improve the model performance.
3) We provide two efficient solutions, including the exact solution based on ST-CUT method and the approximate solution based on conjugate gradient method.

3. AU recognition based on multi-label learning with missing labels

3.1. Problem analysis

For AU recognition, our goal is to detect all the occurrence AUs from expressional images. Suppose that we have n input images

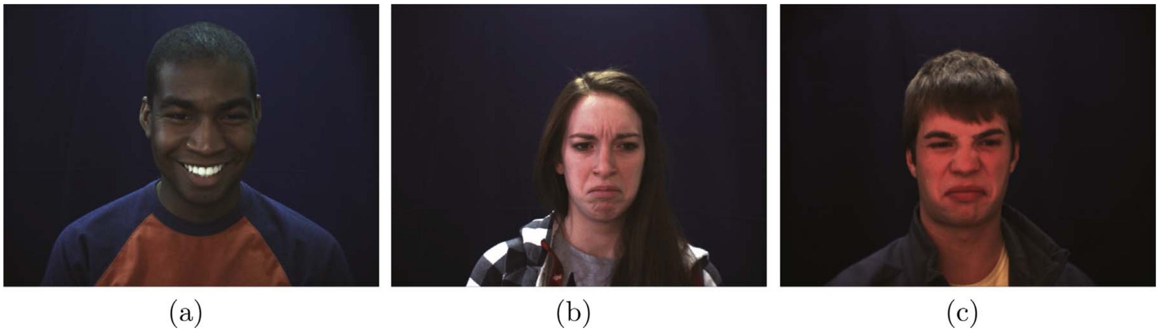


Fig. 2. AU occurs simultaneously to show meaningful expression. (a) AU6+AU12+AU25 to represent happiness. (b) AU4+AU15+AU17 to represent sadness. (c) AU9+AU17 to represent disgust (adapted from [6]).

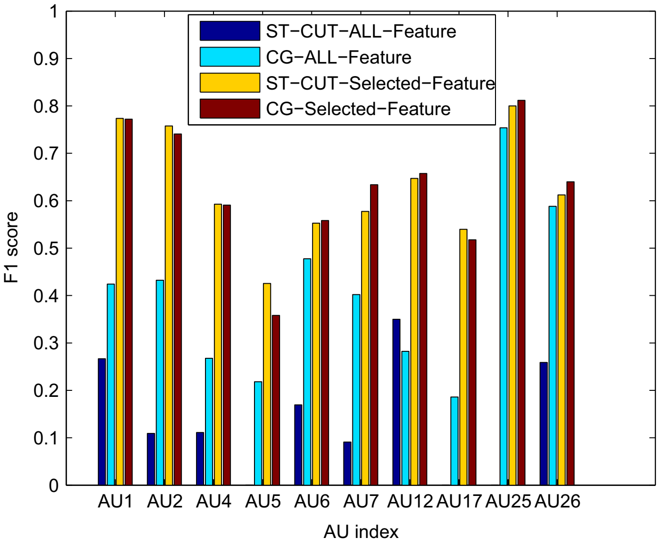


Fig. 3. Evaluation results of ST-CUT-ALL-Feature, CG-ALL-Feature, ST-CUT-Selected-Feature, and CG-Selected-Feature on SEMIANE database for the case of data proportion is 100%.

represented as $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$, where each image is represented as a d-dimensional feature vector. For one expressional image, there may exist multi AUs. For instance, multi AUs usually occur together to express meaningful expressions, e.g., AU6, AU12 and AU25 are present simultaneously to represent happiness (Fig. 4(a)), AU4, AU15 and AU17 happen together to show sadness (Fig. 4(b)), AU9 and AU17 are an illustration of disgust (Fig. 4(c)). The label matrix hence can be expressed as $Y = [y_1, \ldots, y_n] \in \{-1, 0, 1\}^{m \times n}$, which aggregates the m-dimensional label vectors of the instances, and m is the total number of AUs we are going to recognize. Therefore, each image x_i can contain one or more classes from the m different classes $\{c_1, \ldots, c_m\}$, and corresponds to one label vector $y_i = Y_{.i}$ where $Y_{ji} = +1$ means data instance x_i contains the class of c_j and $Y_{ji} = -1$ means x_i does not contain this class. If $Y_{ji} = 0$, then the label c_j of x_i is considered missing, i.e., it has a missing label. With this notation, all m labels of each testing instance x_k are missing.

The objective of AU recognition in this work is to learn a model based on the data X and the provided labels Y, to predict the labels of unlabeled testing data. For this purpose, we are going to obtain a complete label matrix $Z \in \{+1, -1\}^{m \times n}$ that satisfies the label consistency, class-specific instance-level label smoothness, as well as class-level label smoothness. For label consistency, we constrain $Z_{ij} = Y_{ij}$ if $Y_{ij} \neq 0$. Class-specific instance-level smoothness means that if the features for a certain class of two images are similar, then their labels of the corresponding class should be close. Class-level label smoothness means if two classes c_k and c_l have strong semantic meaning, then their instantiations in the overall data set X, represented by the corresponding rows of matrix $Z_{k\cdot}$ and $Z_{l\cdot}$, should also be similar. In the following, we are going to detailedly present the modeling of the label consistency, the modeling of class-specific instance-level label smoothness and the modeling of class-level label smoothness respectively.

3.2. Label consistency modeling

The label consistency modeling of Z with Y is to enforce Z_{ij} to be +1 when $Y_{ij} = +1$, and Z_{ij} is encouraged to be −1 when $Y_{ij} = -1$; when $Y_{ij} = 0$, there is no constraint on Z_{ij}. Hence, we model the consistency of Z with Y as follows:

$$\ell(Y, Z) = \sum_{i,j}^{m,n} W_Y(i, j)(Y_{ij} - Z_{ij}) = const - tr(W_Y^T Z) \tag{1}$$

where $W_Y \in \mathbb{R}^{m \times n}$ is the weight matrix and is defined as:

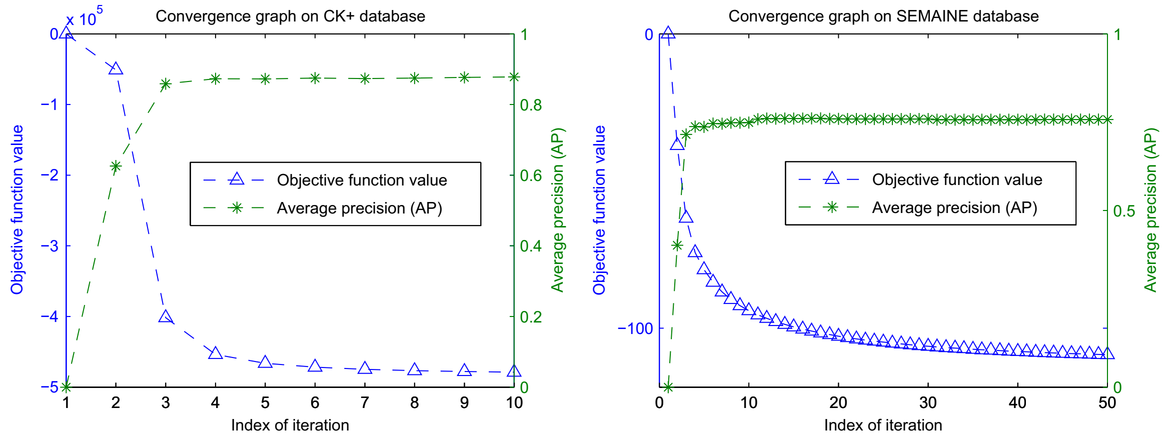
$$W_Y(i, j) = \begin{cases} \dfrac{n(Y_{i\cdot} = -1)}{n(Y_{i\cdot} = +1)} & Y(i, j) = +1 \\ Y(i, j) & Y(i, j) \neq +1 \end{cases} \tag{2}$$

where $n(Y_{i\cdot} = -1)$ is the total number of negative samples for the ith class in Y, and $n(Y_{i\cdot} = +1)$ represents the total number of positive samples for the ith class in Y. Setting W_Y this way embeds the observation that for AU recognition most data instances are with negative labels and positive samples are rare. Hence a higher penalty is incurred if a ground truth label is +1 and predicted as −1.

3.3. Class-specific instance-level label smoothness modeling

Instance-level label smoothness enforces the labels of two data instances to be close if the features of the corresponding two images are similar. Different from previous works, i.e., [8,9], which model the feature-level similarity for all classes based on the same features, we compute the feature similarity over each pair of data instance for each class individually. This is mainly because of the fact that different AUs are caused by the contraction of different face muscles, and hence the occurrence of different AUs can produce changes of skin surface displacement, or face appearance in different face regions. Hence it is obviously unreasonable to use the similarity based on all features of two instances as the common label similarity for all classes.

Sequentially, in this work we define the feature similarity of two data instances w.r.t. one specific AU class k based on the most



(a) (b)

Fig. 4. Convergence graph of *CG-Selected-Feature* on both CK+ and SEMAINE databases.

related features as follows:

$$W_X^k(i,j)=\begin{cases}\exp\left(\dfrac{-\|x_i^k-x_j^k\|^2}{\varepsilon_i{}^k\varepsilon_j^k}\right) & i\neq j,\\ 0 & i=j,\end{cases}\tag{3}$$

where $k\in\{1,2,\ldots,m\}$, $i,j\in\{1,2,\ldots,n\}$, and x^k is the most related feature selected for class k (the kth AU we are going to recognize). We employ supervised learning method, i.e., linear discriminant analysis (LDA) [53], to extract features x^k based on the available training data. For inference, the learned eigenvectors are further applied to the original features. Note that LDA is employed for feature selection, but not for classification.

Based on (3), the class-specific instance-level label smoothness modeling is formulated as follows:

$$S_X(Z)=\sum_k^m\sum_{ij}^n\frac{W_X^k(i,j)}{2}\left(\frac{Z_{ki}}{(d_X^k(i))^{\frac{1}{2}}}-\frac{Z_{kj}}{(d_X^k(j))^{\frac{1}{2}}}\right)^2=\sum_k^m \mathrm{tr}(Z_{k\cdot}L_X^kZ_{k\cdot}^\top),\tag{4}$$

where $Z_{k\cdot}=Z(k,:)$, and the normalized Laplacian matrix $L_X^k=I_n-(D_X^k)^{-\frac{1}{2}}W_X^k(D_X^k)^{-\frac{1}{2}}\in\mathbb{R}^{n\times n}$ with $D_X^k=\mathrm{diag}(d_X^k)$ and $d_X^k(i)=\sum_j W_X^k(i,j)$. Furthermore, we can transform Eq. (4) to the equivalent formulation w.r.t. the vector variable $z=\mathrm{vec}(Z^\top)\in\mathbb{R}^{mn\times 1}$,

$$S_X(z)=\frac{1}{2}\sum_{ij}^{mn}W_X(i,j)\left(\frac{z_i}{\sqrt{d_X(i)}}-\frac{z_j}{\sqrt{d_X(j)}}\right)^2=\mathrm{tr}(z^\top L_Xz),\tag{5}$$

where $W_X\in\mathbb{R}^{mn\times mn}$ with $W_X(i+(k-1)n,j+(k-1)n)=W_X^k(i,j)$, $i,j\in\{1,2,\ldots,n\}, k\in\{1,2,\ldots,m\}$, while all other entries being 0. In addition, we have $L_X=I_{mn}-(D_X)^{-\frac{1}{2}}W_X(D_X)^{-\frac{1}{2}}\in\mathbb{R}^{mn\times mn}$ with $D_X=\mathrm{diag}(d_X)$ and $d_X(i)=\sum_j W_X(i,j)$.

3.4. Class-level label smoothness modeling

For class-level label smoothness modeling, we are going to model the semantic relationships among AUs, which have been proven helpful for improving AU recognition performance [14,16]. As described in FACS manual [5], the inherent relationships among AUs can provide required information to better analyze the facial expressions. One most important inherent relationship among AUs is the co-occurrence relationship. As demonstrated in Fig. 2, the co-occurrence relationship characterizes the groups of AUs, which oftentimes appear together to show meaningful facial emotions. Such co-occurrence relationship is embedded in the labels and is often referred as *prior information*. In this work, we first define an $m\times m$ nonnegative matrix w_C to measure the co-occurrence relationship between classes as:

$$\overline{W}_C(i,j)=\begin{cases}\dfrac{<\overline{Y}_{i\cdot},\overline{Y}_{j\cdot}>}{\|\overline{Y}_{i\cdot}\|\cdot\|\overline{Y}_{j\cdot}\|} & i\neq j\\ 0 & i=j,\end{cases}\tag{6}$$

where $\overline{Y}_{i\cdot}$ is a binary n-dimensional row vector with the entries corresponding to the positive labels in $Y_{i\cdot}$ being 1, while all other entries being 0. Then we model the class-level label smoothness as follows:

$$S_C(Z)=\sum_k^n\sum_{ij}^m\frac{\overline{W}_C(i,j)}{2}\left(\frac{Z_{ik}}{(\overline{d}_C(i))^{\frac{1}{2}}}-\frac{Z_{jk}}{(\overline{d}_C(j))^{\frac{1}{2}}}\right)^2=\mathrm{tr}(Z^\top\overline{L}_CZ),\tag{7}$$

where the corresponding normalized Laplacian matrix $\overline{L}_C=I_m-(\overline{D}_C)^{-\frac{1}{2}}\overline{W}_C(\overline{D}_C)^{-\frac{1}{2}}\in\mathbb{R}^{m\times m}$ with $\overline{D}_C=\mathrm{diag}(\overline{d}_C)$, where $\overline{d}_C(i)=\sum_j^m\overline{W}_C(i,j)$. Furthermore, we can transform Eq. (7) to the equivalent formulation w.r.t. the vector variable z as:

$$S_C(z)=\frac{1}{2}\sum_{ij}^{mn}W_C(i,j)\left(\frac{z_i}{\sqrt{d_C(i)}}-\frac{z_j}{\sqrt{d_C(j)}}\right)^2=\mathrm{tr}(z^\top L_Cz),\tag{8}$$

where $W_C=\overline{W}_C\otimes I_n\in\mathbb{R}^{mn\times mn}$, where ⊗ indicates the Kronecker product [19]. $L_C=I_{mn}-(D_C)^{-\frac{1}{2}}W_C(D_C)^{-\frac{1}{2}}\in\mathbb{R}^{mn\times mn}$ with $D_C=\mathrm{diag}(d_C)$, where $d_C(i)=\sum_j^{mn}W_C(i,j)$.

3.5. Objective function

We formulate the AU recognition based on MLML as a binary optimization problem by linearly combining Eqs. (1), (5), and (8), as follows:

$$\arg\min_{z\in\{-1,+1\}^{mn\times 1}}\ -w_Y^\top z+\beta z^\top L_Xz+\gamma z^\top L_Cz=-w_Y^\top z+z^\top Lz\tag{9}$$

where $w_Y=\mathrm{vec}(W_Y^\top)\in\mathbb{R}^{mn\times 1}$, and $L=\beta\cdot L_X+\gamma\cdot L_C$. The two pre-set parameters β and γ control the impact of the feature-level similarity term and the class-level dependency term, and can be tuned by cross validation.

3.6. Optimization

3.6.1. Discrete optimization by ST-CUT method

Definition 1. (Submodular function, see Definition 2.1 in [20]) A set-function F: $2^V\to\mathbb{R}$ is submodular if and only if, for all subsets $A,B\subseteq V$, we have: $F(A)+F(B)\geq F(A\cup B)+F(A\cap B)$. It can also be equivalently represented via indicator vector $\mathbf{1}_A\in\{-1,1\}^{|V|}$: if the point i belongs to A, then $\mathbf{1}_A(i) = 1$, otherwise $\mathbf{1}_A(i) = -1$. Then the sufficient and necessary condition of submodular function can be reformulated as $F(\mathbf{1}_A) + F(\mathbf{1}_B) \geq F(\mathbf{1}_{A\cup B}) + F(\mathbf{1}_{A\cap B})$.

Definition 2. (Modular function, see Proposition 2.1 in [20]) A set-function $F: 2^V \to \mathbb{R}$ is modular if and only if there exists $s \in \mathbb{R}^{|V|}$ such that $F(A) = \sum_{i\in A} s_i$, i.e., $F(\mathbf{1}_A) = s^\top \mathbf{1}_A$. Modular function is also submodular, and it plays for set-functions the same role as linear functions for continuous functions.

Lemma 1. *[20] According to Definition 1, it is clear that the positive linear combination of submodular functions is still submodular.*

Proposition 1. *Problem (9) can be globally solved by ST-CUT algorithm [56].*

Proof. Firstly we define a set function $F_{ijk}: \{-1, 1\}^2 \to R$ between two connected label nodes Z_{ki} and Z_{kj}. According to the definition in Eq. (7), the class correlation between them can be represented as $F_{ijk}(Z_{ik}, Z_{jk}) = W_C(i, j)\left(Z_{ik}/\sqrt{d_C(i)} - Z_{jk}/\sqrt{d_C(j)}\right)^2$. It is easy to obtain that $F(-1, -1) + F(1, 1) < F(-1, 1) + F(1, -1)$. It means that when two labels are positively correlated (i.e., $W_C(i, j) > 0$), then the cost that this two labels are same is lower than the cost that they are different. According to Definition 1, we know that F is a submodular function. Moreover, $S_C(Z) = \frac{1}{2}\sum_k^n \sum_{i,j}^m F(Z_{ik}, Z_{jk})$ can be seen as a positive linear combination of a set of submodular functions. According to Lemma 1, $S_C(Z)$ is also submodular. Similarly, it is easy to know that $S_X(Z)$ is submodular, as it is also a positive linear combination of a set of submodular functions. Moreover, the consistency term $\ell(Y, Z)$ in Eq. (1) is a linear function of Z_{ij}. According to Definition 2, $\ell(Y, Z)$ is a modular function, and is also submodular. Thus, since the objective function (9) is a positive linear combination of two submodular and one modular functions, it is submodular.

Interestingly Problem (9) has the same formulation with the standard graph-based semi-supervised learning [57]. It can be seen as a graph cut problem, where the linear term $-w_Y^\top z$ serves as the unary potential for single nodes, while the quadratic term $z^\top Lz$ is considered as the summation of the edge potentials. As demonstrated in [56], the graph cut problem with submodular function can be globally solved by ST-CUT algorithm. □

3.6.2. Continuous optimization by conjugate gradient method

Although the ST-CUT algorithm gives the global minimum of Problem (9), it does not mean its performance is the best. There are two main reasons. Firstly, as demonstrated in [57], due to the existence of multiple global minimums, ST-CUT algorithm may output the biased results (e.g., most labels are predicted as negative). Secondly, the loss function used in (9) (i.e., weighted Hamming loss) and the evaluation metric for multi-label prediction could be inconsistent.

Thus we also present a continuous optimization algorithm, through relaxing the original discrete problem (9) into the following continuous problem,

$$\arg\min_z -w_Y^T z + z^T Lz, \quad s.t.\ z \in [-1, +1]^{mn\times 1}. \tag{10}$$

It is easy to prove Problem (10) is a convex minimization problem as shown in Proposition 2.

Lemma 2. *[60] L is positive semi-definite (PSD).*

Proof. Since L_X and L_C are all normalized Laplacian matrix, L_X and L_C are all positive semi-definite (PSD) matrices [58]. Hence, as L is the positive weighted linear combination of two PSD matrices, it is easy to conclude that L is a PSD matrix. □

Proposition 2. *Problem (10) is convex minimization.*

Proof. The Hessian of the objective function in (10) w.r.t. z is L, which has been proven to be a PSD matrix in Lemma 2. Thus the objective function (10) is a convex function in z. The box constraints lead to a convex feasible solution space, so it is easy to conclude that Problem (10) is a convex minimization problem. Besides, obviously the box constraint $[0, 1]$ is strictly feasible, as $(0, 1)$ is its subspace. According to [59], Problem (10) satisfies Slater's condition. Moreover, since Problem (10) is convex, then the strong duality holds, i.e., the duality gap is 0. □

Since Problem (10) is convex, many off-the-shelf optimization methods can be adopted to globally optimize it. Here we utilize the *conjugate gradient* (CG) descent algorithm [55]. In each iteration we add a projection step that projects the updated solution into the box space $[-1, 1]$. CG method is often implemented as an iterative algorithm, applicable to sparse systems that are too large to be handled by a direct implementation or other direct methods. For CG algorithm, the search directions are constructed by conjugation of the residuals, and each new residual is orthogonal to all the previous residuals and search directions.

4. Experiments

In this section, we evaluate the proposed method on both posed facial expression database and spontaneous facial expression database. For posed expression database, the extend Cohn–Kanade (CK+) database [24] is adopted. CK+ database contains 593 posed facial activity videos from 210 adults, among which, 9% are female, 81% are Euro-American, 13% are Afro-American and 6% are from other groups. CK+ database has been widely used for evaluating facial activity recognition system, and one advantage of using CK+ database is that this database demonstrates diversity over subjects and it involves multiple-AU expressions. Besides, we are also going to evaluate the proposed method on spontaneous facial expression database, and the SEMAINE database [25] is adopted for this purpose. Unlike CK+ database, the expressions of users in SEMAINE are naturally induced by operators during the conversation. Therefore the database contains speech related mouth and face movements, and significant amounts of both in- and out-of-plane head rotations. All these issues make the recognition task much more challenging.

4.1. Experiment settings

Evaluation metrics: Since AU data is usually unbalanced and the positive sample proportion is usually low, we use F1 score as the performance metric which is computed as:

$$F1 = 2\frac{P \times R}{P + R} \tag{11}$$

where P is the precision and R is the recall. Besides, another widely used evaluation metric for multi-label ranking, namely, average precision (AP) [54] is also adopted. Specifically, AP is calculated as follows [54]:

$$AP = \frac{1}{n}\sum_i^n \frac{1}{S^i}\sum_{s_r}^{S^i} \frac{|\{s_t \in S^i rank(x_i, s_t) < rank(x_i, s_r)\}}{rank(x_i, s_r)} \tag{12}$$

where S^i denotes the set of ground-truth positive classes of sample x_i, and $rank(x_i, s_t)$ represents the rank order of class s_t in the label ranking list of x_i. Higher values of F1 and AP represent better performances. Note that F1 score is computed based on binary predicted labels while AP is the overall evaluation of continuous label ranking list and the AP score corresponding to each AU is not calculated.

Comparisons: We are going to make a detailed comparison between discrete optimization method, i.e., ST-CUT, and

Table 2
Selecting feature V.S. All feature for CG algorithm (Evaluation on CK+ database using leave-one-subject-out cross validation strategy. Digital values of the entries are the F1 scores for each AU, and Avg. represents the average F1 score of all the 16 target AUs.).

Data	Methods	AU1	AU2	AU4	AU5	AU6	AU7	AU9	AU12	AU14	AU15	AU17	AU20	AU23	AU24	AU25	AU27	Avg.
5%	CG-ALL-Feature	64.1	70.8	50.0	61.2	63.6	46.4	69.1	64.7	34.5	36.6	51.2	41.5	43.5	36.6	50.8	67.2	53.2
	CG-Selected-Feature	83.7	80.0	74.6	77.2	68.0	59.4	70.9	77.2	33.6	39.4	68.2	36.9	50.0	50.0	86.0	84.0	**64.9**
10%	CG-ALL-Feature	80.6	80.0	69.1	66.3	74.3	52.4	64.2	80.9	24.1	31.5	74.7	26.7	49.2	40.9	92.2	89.6	62.3
	CG-Selected-Feature	90.0	88.7	78.2	82.7	74.2	59.0	86.9	83.5	43.9	54.8	81.2	57.7	65.3	51.6	95.2	90.3	**73.9**
20%	CG-ALL-Feature	81.1	86.2	75.5	74.3	75.9	58.6	71.4	81.5	37.3	37.5	79.8	36.1	42.1	42.9	92.4	90.4	66.4
	CG-Selected-Feature	90.0	87.6	79.2	81.1	73.3	60.0	86.9	85.7	45.9	54.8	81.5	53.9	63.9	55.1	95.2	89.7	**73.9**
50%	CG-ALL-Feature	80.3	88.0	75.1	79.1	77.0	56.1	72.4	87.7	35.3	33.8	83.1	35.9	37.7	41.3	92.4	90.5	66.6
	CG-Selected-Feature	90.8	89.1	77.7	82.7	73.7	58.9	85.9	84.5	42.9	55.6	83.4	52.8	65.9	55.1	96.1	90.3	**74.1**
100%	CG-ALL-Feature	75.2	85.1	75.6	78.8	72.9	54.4	75.2	80.3	51.4	42.7	83.9	40.0	52.3	48.3	94.3	90.0	68.8
	CG-Selected-Feature	91.2	89.1	78.7	83.2	73.3	60.5	85.9	85.0	43.9	56.3	82.9	56.1	63.9	53.6	95.8	90.3	**74.4**

continuous optimization method, i.e., CG, in the experiment part. Moreover, detailed comparison and analysis between selecting features and using all features for both ST-CUT and CG are provided. We denote the corresponding methods as *ST-CUT-Selected-Feature*, *ST-CUT-ALL-Feature*, *CG-Selected-Feature* and *CG-ALL-Feature*, respectively. For comparing selecting features and using all features, all the parameter settings are the same but *-Selected-Feature* uses features individually selected for each AU while *-ALL-Feature* uses the same features for all classes. For comparison with other works, several state-of-the-art multi-label learning methods that can handle missing labels are used for comparison, including SMSE2 [13], MLR-GL [22], MC-Pos [23] and MLML-exact [8]. We implement SMSE2 and MLML-exact in Matlab and adopt the publicly available code for MLR-GL. The code of MC-Pos is provided by its authors. We make our best effort to adjust the parameters in these methods as suggested in the original papers. For all experiments, we employ leave-one-subject-out cross validation strategy.

Other settings: To simulate different scenarios with missing labels, we create training datasets with varying portions of provided labels, i.e., label proportion of 5% means 95% of the whole training label matrix is missing while label proportion of 100% represents no missing labels. In each case, the missing labels are randomly chosen and set to 0 in the original label matrix of the training data set. The trade-off parameters β and γ are tuned by two-step cross-validation. The optimal values of β and γ depend on data, and in this work $\beta = 1, \gamma = 0.01$ on CK+ database, and $\beta = 5, \gamma = 0.01$ on SEMAINE database.

4.2. Evaluation results on posed expression database

We collect the peak frames of 327 sequences from CK+ database, which are provided with both expression and 16 most frequent AU labels. The 16 AUs we are going to recognize are AU1, 2, 4, 5, 6, 7, 9, 12, 14, 15, 17, 20, 23, 24, 25, and 27. Each image is described by a 201-dimensional column vector, which is cascaded by four types of features: 102-dimensional vector of the location of 51 facial feature points; 40-dimensional texture features; 30-dimensional appearance features; 29-dimensional shape features. We apply principal component analysis (PCA) to the original features to reduce the redundant information, and the selected components remain at least 90% of the information. The selected features by PCA are used as the features for *ST-CUT-ALL-Feature* and *CG-ALL-Feature*. We further employ supervised feature selecting method, i.e., linear discriminant analysis (LDA) [53], to extract features for each AU individually, and the extracted features are fed to *ST-CUT-Selected-Feature* and *CG-Selected-Feature*.

Selecting feature V.S. all feature:

Table 2 demonstrates the evaluation results of *CG-Selected-Feature* and *CG-ALL-Feature* on CK+ database respectively. From Table 2 we can see that *CG-Selected-Feature* significantly outperforms *CG-ALL-Feature* for all label proportion cases. Furthermore, the performance improvement *CG-Selected-Feature* achieved over *CG-ALL-Feature* increases as the missing label proportion increases. For instance, for the cases of label proportion is 100%, 50%, 20%, 10% and 5%, *CG-Selected-Feature* improves the average F1 score of *CG-ALL-Feature* by 5.6%, 7.5%, 7.5%, 11.6% and 11.7% respectively. This is mainly because that as the missing labels increase, the effect of label consistency term weakens and correspondingly the effect of feature similarity term strengthens. The feature similarity calculated by *CG-Selected-Feature* is more precise than that of *CG-ALL-Feature*, hence the improvement of *CG-Selected-Feature* compared to *CG-ALL-Feature* increases as the effect of feature similarity term increases. Table 2 also demonstrates that for AUs that produce subtle feature changes the improvements of *CG-Selected-Feature* compared to *CG-ALL-Feature* are much more significant. For instance, the occurrence of AU15, AU20 and AU23 all produce subtle feature changes compared to other lower face AUs, and *CG-Selected-Feature* improves the F1 score of *CG-ALL-Feature* for these three AUs by 13.6%, 16.1% and 11.6% for the case of label proportion is 100%, respectively. The possible reason for this is that for AUs that produce subtle feature changes, the discriminative information for these AUs is kind of submerged by other feature changes when using all features. That is why *CG-ALL-Feature* gets poor performances on these AUs.

Table 3 lists the evaluation results of ST-CUT using all features and selected features on CK+ database respectively. From Table 3 we can see that similar as previous case for CG method, *ST-CUT-Selected-Feature* outperforms *ST-CUT-ALL-Feature* significantly for all cases. For instance, *ST-CUT-Selected-Feature* improves the average F1 score of *ST-CUT-ALL-Feature* by 14.9% for the case of label proportion is 100%. Also, for AUs that produce subtle feature changes such as AU15, AU20 and AU23, the improvements of *ST-CUT-Selected-Feature* over *ST-CUT-ALL-Feature* are much more significant. For example, *ST-CUT-Selected-Feature* improves the F1 score of *ST-CUT-ALL-Feature* for AU15, AU20 and AU23 by 48.2%, 46.0% and 31.3% for the case of label proportion is 100%, respectively. From Table 3 we can also see that when there are missing labels the F1 scores of some AUs for *ST-CUT-ALL-Feature* are *NaN*, which may be due to that *ST-CUT-ALL-Feature* outputs biased

Table 3
Selecting feature V.S. All feature for ST-CUT algorithm (Evaluation on CK+ database using leave-one-subject-out cross validation strategy. Digital values of the entries are the F1 scores for each AU, and Avg. represents the average F1 score of all the 16 target AUs.).

Data	Methods	AU1	AU2	AU4	AU5	AU6	AU7	AU9	AU12	AU14	AU15	AU17	AU20	AU23	AU24	AU25	AU27	Avg.
5%	ST-CUT-ALL-Feature	17.9	11.0	9.5	14.8	6.6	17.0	3.1	12.2	NaN	NaN	8.9	NaN	NaN	NaN	30.3	4.8	NaN
	ST-CUT-Selected-Feature	80.4	80.6	71.5	71.8	60.7	50.2	81.4	70.1	31.4	50.0	60.1	40.3	51.1	40.7	81.8	81.6	**62.7**
10%	ST-CUT-ALL-Feature	NaN	18.4	NaN	10.1	NaN	NaN	NaN	2.5	NaN	NaN	NaN	NaN	NaN	NaN	2.2	19.5	NaN
	ST-CUT-Selected-Feature	88.4	88.2	79.7	80.9	73.6	57.8	89.4	81.4	41.5	50.0	79.5	52.0	59.2	55.7	90.3	87.7	**72.2**
20%	ST-CUT-ALL-Feature	59.5	85.1	7.8	75.3	56.1	2.5	NaN	71.8	NaN	NaN	23.7	NaN	NaN	NaN	2.2	90.4	NaN
	ST-CUT-Selected-Feature	87.9	87.3	77.7	79.8	73.8	62.4	89.4	84.0	43.4	51.9	82.4	63.3	60.6	53.1	95.2	90.4	**73.9**
50%	ST-CUT-ALL-Feature	73.7	87.6	76.6	77.9	77.4	51.1	66.7	87.6	NaN	NaN	81.9	NaN	NaN	NaN	94.2	91.9	Nan
	ST-CUT-Selected-Feature	90.0	88.1	78.1	82.9	72.9	60.7	89.4	83.7	41.3	56.0	81.2	59.3	64.0	54.9	94.0	90.4	**74.2**
100%	ST-CUT-ALL-Feature	79.2	87.9	76.0	79.6	76.0	54.4	75.2	86.3	5.6	5.9	82.9	13.3	32.7	7.8	94.6	93.4	59.4
	ST-CUT-Selected-Feature	91.7	87.7	76.9	82.5	73.2	59.9	89.4	83.7	43.0	54.1	81.3	59.3	64.0	55.5	95.6	90.5	**74.3**

results and predicts all the labels as negative. The possible reason for the biased output is the existence of multiple global minimums, and when there is a certain amount of potential solutions ST-CUT method tends to output biased results [57]. Another reason for the bad performance of ST-CUT method is the inconsistency between the loss function, i.e., weighted Hamming loss, and the evaluation metric, i.e., F1 score.

ST-CUT V.S. CG: Tables 2 and 3 list the results for *CG-Selected-Feature* and *ST-CUT-Selected-Feature*. From Tables 2 and 3 we can see that for the cases of label proportion is 100%, 50% and 20%, *CG-Selected-Feature* and *ST-CUT-Selected-Feature* achieve similar results. However, for the cases of label proportion is 10% and 5%, *CG-Selected-Feature* improves the average F1 score of *ST-CUT-Selected-Feature* by 1.7% and 2.2% respectively. There is a trend that the lower the label proportion is the more improvement CG achieved compared to ST-CUT. To further demonstrate this point, we tested an extreme case that there is only 1% label in the training data. For this case *CG-Selected-Feature* achieves an average F1 score of 58.0% while *ST-CUT-Selected-Feature* achieves an average F1 score of 43.4%. There is a 14.6% improvement on average F1 score for this case. This is mainly because that as the missing labels increase, for ST-CUT method the number of links connected to *s* node and *t* node decreases and hence the number of potential global solutions increases. As discussed before, ST-CUT tends to output bias results when there is a certain amount of global solutions [57]. This is why *ST-CUT-Selected-Feature* gets poor performance when the amount of missing labels increases. Another advantage of continuous optimization method, i.e., CG, is the continuous output, which can be further transformed to the intensity or classification confidence of the predicted results.

Comparison with related works: For comparison with other works, several state-of-the-art multi-label learning methods that can handle missing labels are adopted, including SMSE2 [13], MLR-GL [22], MC-Pos [23] and MLML-exact [8]. The evaluation results on CK+ database are shown in Table 4. From Table 4 we can see that *CG-Selected-Feature* and *ST-CUT-Selected-Feature* outperform compared methods for most cases. For the case of label proportion is 100%, the average F1 scores of *CG-Selected-Feature* and *ST-CUT-Selected-Feature* are slightly worse than that of MC-Pos but better than that of other works. Furthermore, when there are missing labels the average F1 scores of *CG-Selected-Feature* and *ST-CUT-Selected-Feature* are significantly better than that of other works. For the evaluation metric AP, the scenario is similar. For instance, for the case of label proportion is 100%, the AP of *CG-Selected-Feature* is slightly worse than that of MC-Pos but better than other works. However for the case of label proportion is 5%, *CG-Selected-Feature* improves the AP of MLR-GL, MC-Pos, and MLML-exact by 8.2%, 7.8% and 23.4% respectively. This may be due to two main reasons. One possible reason is that when there are missing labels, SMSE2, MLR-GL and MC-Pos involve many noisy negative labels in the original training label matrix, i.e., some positive labels 1 are set to 0. However, it is not the case for *CG-Selected-Feature* and *ST-CUT-Selected-Feature* since the proposed methods formulate the learning with missing labels as a semi-supervised learning model and do not give any bias to missing labels. In contrast, missing labels are encouraged to be intermediate values between negative and positive labels in SMSE2, MLR-GL and MC-Pos , which brings in label bias. This is why their performance decreases significantly as the missing proportion increases. Another reason is that all the compared methods employ the global features and hence involve much noise from the occurrences of other AUs, while the proposed method only use the most related features to discriminate each AU. As the proportion of missing labels increases, the effect and label consistency term weakens and correspondingly the effect of feature similarity term strengthens. Since the class-specific feature similarity term is more precise, the superiority of the proposed methods is more remarkable as the amount of missing labels increases.

Table 4
Comparison with other works (evaluation on CK+ database using leave-one-subject-out cross validation strategy).

Methods	F1					AP				
	5%	10%	20%	50%	100%	5%	10%	20%	50%	100%
SMSE2 [13]	–	–	66.0	73.3	74.2	–	–	78.8	84.4	87.3
MLR-GL [22]	55.5	62.5	68.1	71.6	73.8	67.6	75.9	79.8	83.9	86.5
MC-Pos [23]	56.3	62.3	67.0	73.8	**75.2**	67.9	78.5	81.1	85.8	**88.1**
MLML-exact [8]	49.2	53.6	61.5	69.3	73.0	52.4	65.7	73.4	81.5	86.1
ST-CUT-Selected-Feature	62.7	72.2	73.9	74.2	74.3	–	–	–	–	–
CG-Selected-Feature	**65.0**	**74.0**	**74.0**	**74.1**	74.4	**75.8**	**82.5**	**84.1**	**85.9**	87.4

4.3. Evaluation results on spontaneous expression database

To evaluate the proposed method on spontaneous expression database, we collect a dataset from the SEMAINE database [25]. The SEMAINE database consists of a large number of emotionally colored conversations, and all the expressions of users are naturally induced by operators during the conversation [25]. The SEMAINE database includes speech related mouth and face movements, and significant amounts of both in- and out-of-plane head rations, which make the recognizing task more challenging [48]. So far a total of 180 frames from 8 sessions of SEMAINE are FACS annotated by experts, and in this experiment we recognize 10 AUs that are present for at least 15 times. The target AUs are AU1, 2, 4, 5, 6, 7 ,12, 17, 25, 26.

Following the work in [48,49], we use the Local Binary Pattern (LBP) feature in this part of experiment. The LBP feature is extracted in the same manner as [48]. Similar as previous case on CK+ database, PCA is applied to the original features and the selected components remain at least 90% of the information. The selected features by PCA are used as the features for *ST-CUT-ALL-Feature* and *CG-ALL-Feature*. We then further employ LDA to select features for each AU individually and the selected features are fed to *ST-CUT-Selected-Feature* and *CG-Selected-Feature*. The evaluation results of *ST-CUT-ALL-Feature*, *CG-ALL-Feature*, *ST-CUT-Selected-Feature* and *CG-Selected-Feature* for the case of data proportion is 100% are demonstrated in Fig. 3.

From Fig. 3 we can see that individually selecting features for each AU significantly outperforms using same features for all AUs. For instance, *ST-CUT-Selected-Feature* and *CG-Selected-Feature* achieve an average F1 score of 62.8% and 62.8% respectively, while *CG-ALL-Feature* achieves an average F1 score of 40.3%. The improvement on average F1 score is over 20% for *CG* method. The performance of *ST-CUT-ALL-Feature* is even worse, e.g., the F1 scores of *ST-CUT-ALL-Feature* for AU5, AU17 and AU25 are *NaN* because *ST-CUT-ALL-Feature* outputted biased results (all negative) for these AUs. From Fig. 3 we can also see that the performance gap between *CG-ALL-Feature* and *CG-Selected-Feature* for AU25 and AU26 is relatively much smaller than that for other AUs. For example, *CG-Selected-Feature* outperforms *CG-ALL-Feature* by 5.8% and 5.5% (F1 score) for AU25 and AU26, while the improvement of average F1 score is 22.5%. This is mainly because that the occurrences of AU25 and AU26 produce more significant feature changes compared to other AUs. Hence discriminative information for these two AUs is kind of not badly influenced by feature changes of other AUs. In contrast, the occurrence of AU17 produces subtle feature changes compared to other lower face AUs, and hence using all features will involve much noise for this AUs. For instance, the F1 score of *CG-ALL-Feature* for AU17 is 18.6%, while *CG-Selected-Feature* achieves an F1 score of 51.8% for this AU.

Table 5 lists the comparison with related works on SEMAINE database. From Table 5 we can find that, the proposed methods constantly outperform the compared methods. For instance, for the case of data proportion is 100% *ST-CUT-Selected-Feature* and *CG-Selected-Feature* achieve an average F1 score of 62.8% and 62.8% respectively compared to 53.4% for SMSE2, 59.2% for MLR-GL, 59.9% for MC-Pos, and 50.1% for MLML-exact. The improvement on SEMAINE database is much more significant compared to that on CK+ database. This may be due to that the features extracted on SEMAINE database contain much more noise because of the in- and out-of-plane head ratios. The compared methods all employ the global features to discriminate all the classes, and hence involve most noise of the features. Nevertheless, for the proposed methods, only the most related features for each AU are used and hence calculating the class-specific feature similarity term avoids much noise from head rations and occurrences of other AUs. That is why the performances of the proposed methods are better than that of the compared methods on SEMAINE database.

From Table 5 we can also see that similar as previous case on CK+ database, when there are missing data, the improvement of the proposed methods compared to the comparison works are more significant. For example, for the case of label proportion is 100% *CG-Selected-Feature* improves the AP of MLR-GL, MC-Pos and MLML-exact by 3.6%, 2.6% and 9.7% respectively, while for the case of label proportion is 5% *CG-Selected-Feature* improves the AP of MLR-GL, MC-Pos and MLML-exact by 11.3%, 14.3% and 22.8% respectively. As discussed before, this is mainly due to two reasons, i.e., involving much noise by setting missing labels as negative labels (MLR-GL and MC-Pos) and calculating feature similarity term using all features for all AUs (MLML-exact).

The most recent works by Jiang et al. [48] and Wang et al. [49] represent the state of the art methods for AU recognition and report average F1 scores of 60.8% and 56.1% on SEMAINE database. Note that work [48] recognizes 6 upper face AUs and work [49] recognizes the same 10 AUs as this work. The proposed method *CG-Selected-Feature* achieves an average F1 score of 60.9% for the same 6 upper AUs as in [48], and an average F1 score of 62.8% for the same 10 AUs as in [49], which are similar or better than that of the compared works. Besides, the proposed method can handle missing labels in a principled manner, which is also a significant superiority compared to works [48,49].

Table 5
Comparison with other works (evaluation on SEMIANE database).

Methods	F1					AP				
	5%	10%	20%	50%	100%	5%	10%	20%	50%	100%
SMSE2 [13]	–	–	43.0	46.9	53.4	–	–	59.8	67.6	71.0
MLR-GL [22]	44.4	45.2	47.2	50.8	59.2	58.3	61.0	64.1	65.8	72.2
MC-Pos [23]	43.5	46.2	49.5	52.6	59.9	55.3	58.5	63.5	68.5	73.2
MLML-exact [8]	37.4	40.4	42.7	46.6	50.2	46.8	49.7	53.5	60.9	66.1
ST-CUT-Selected-Feature	54.0	57.9	61.1	61.8	62.8	–	–	–	–	–
CG-Selected-Feature	**56.9**	**59.6**	**61.7**	**61.8**	**62.8**	**69.6**	**73.5**	**75.0**	**76.5**	**75.8**

4.4. Computational complexity and convergence

Table 6 lists the computational complexity of the proposed methods and the compared algorithms. From Table 6 we can see that the computation cost of *ST-CUT-Selected-Feature* is $O(m^2n^2(N_x nm + N_c mn))$ [56], where N_x is the number of neighbors connected to each data instance and N_c is the number of neighbors connected to each class. However, please note that this is the worst case complexity for ST-CUT algorithm. For the proposed method, we set $N_x=20$ and $N_c=4$. Hence the computation cost of *ST-CUT-Selected-Feature* is higher than that of the compared methods. For the proposed algorithm *CG-Selected-Feature*, the overall complexity is $O(T(N_x nm + N_c mn))$ where T is the number of iterations and $(N_x nm + N_c mn)$ is the number of non-zero elements in L (Eq. (10)). The complexity of *CG-Selected-Feature* is comparable

Table 6
Comparison of computational complexities of different algorithms.

Algorithm	SMSE2 [13]	MLR-GL [22]	MC-Pos [23]	MLML-exact [8]	ST-CUT-Selected-Feature	CG-Selected-Feature
Complexity	$O(n^3)$	$O(mn^2)$	$O(mn^2+n^3)$	$O(n^3+m^3)$	$O(m^2n^2(N_x nm + N_c mn))$	$O(T(N_x nm + N_c mn))$

with MLR-GL, but faster than other algorithms. The convergence curves of *CG-Selected-Feature* on both CK+ and SEMAINE databases are shown in Fig. 4. The corresponding AP at each iteration step is also plotted in Fig. 4. From Fig. 4 we can see that *CG-Selected-Feature* converges in a small number of iterations, which guarantees the computational efficiency of the proposed algorithm.

5. Conclusion and future works

In this paper, we proposed a multi-label learning with missing labels (MLML) framework for AU recognition under incomplete data. Different from previous MLML works which usually use the same features for all classes, the proposed method discriminates each AU based on the most related features. Selecting features for each AU individually embeds the observation that occurrences of different AUs produce feature changes of different face regions. Hence using all features (same feature) will obviously involve much noise from occurrences of other AUs, and therefore limit the model performance. Both discrete optimization methods, i.e., ST-CUT algorithm, and continuous optimization method, i.e., CG algorithm, are introduced to solve the formulated learning problem. Sufficient evaluations on both posed and spontaneous databases demonstrate the effectiveness and superiority of the proposed method.

For the further work, we plan to do feature extraction and label propagation alternatively in a unified manner, since the assignment of missing labels through propagation is supposed to be beneficial to extracting better features, and vice versa. Furthermore, there are many real-world problems that can be formulated as multi-label learning with missing labels problem, such as image annotation, text classification, etc. In the future, we aim to apply the proposed method to these applications. Besides, for class-level label smoothness we only model the co-occurrence relationships among labels in this work. Other types of label correlations, i.e., mutual exclusion [14] and semantic hierarchy [10], have been explored in existing works. For the further work, we are going to further combine these label correlations, and the performance of the model is expected to be further improved.

Acknowledgments

Yongqiang Li is supported by National Natural Science Foundation of China (No. 61402129), and Postdoctoral Foundation Projects (Nos. LBH-Z14090 and 2015M571417). Bernard Ghanem and Baoyuan Wu are supported by funding from King Abdullah University of Science and Technology (KAUST). Hongxun Yao is partially supported by National Natural Science Foundation of China (No. 61472103) and Key Program (No. 61133003).

References

[1] A. Savran, B. Sankur, M.T. Bilge, Comparative evaluation of 3D vs. 2D modality for automatic detection of facial action units, Pattern Recognit. 45 (2) (2012) 767–782.
[2] C.L. Breazeal, Sociable machines: expressive social exchange between humans and robots, Diss. Massachusetts Institute of Technology, 2000.
[3] F. Dornaika B. Raducanu, Facial expression recognition for HCI applications, in: Prentice Hall Computer Applications in Electrical Engineering Series, 2009, pp. 625–631.
[4] P. Ekman, Darwin, deception, and facial expression, Ann. New Y. Acad. Sci. 1000 (1) (2003) 205–221.
[5] P. Ekman, W.V. Friesen, J.C. Hager, Facial action coding system, A Human Face, Salt Lake City, UT, 2002.
[6] S.M. Mavadati, M.H. Mahoor, K. Bartlett, P. Trinh, J.F. Cohn, DISFA: a spontaneous facial action intensity database, IEEE Trans. Affect. Comput. 4 (2) (2013) 151–160.
[7] O. Rudovic, V. Pavlovic, M. Pantic, Context-sensitive dynamic ordinal regression for intensity estimation of facial action units, IEEE Trans. Pattern Anal. Mach. Intell. 37 (5) (2015) 944–958.
[8] B. Wu, Z. Liu, S. Wang, B. Hu, Q. Ji, Multi-label learning with missing labels, in: ICPR, IEEE Computer Society, Stockholm, Sweden, 2014, pp. 1964–1968.
[9] B. Wu, S. Lyu, B. Hu, Q. Ji, Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognit. 48 (7) (2015) 2279–2289.
[10] B. Wu, S. Lyu, B. Ghanem, ML-MG: multi-label learning with missing labels using a mixed graph, in: ICCV, IEEE, Santiago, Chile, 2015, pp. 4157–4165.
[12] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.
[13] G. Chen, Y. Song, F. Wang, C. Zhang, Semi-supervised multi-label learning by solving a Sylvester equation, in: SDM, 2008, pp. 410–419.
[14] Y. Tong, W. Liao, Q. Ji, Facial action unit recognition by exploiting their dynamic and semantic relationships, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1683–1699.
[16] Y. Li, J. Chen, Y. Zhao, Q. Ji, Data-free prior model for facial action unit recognition, IEEE Trans. Affect. Comput. 4 (2) (2013) 127–141.
[18] Z. Zeng, M. Pantic, G.I. Roisman, T.S. Huang, A survey of affect recognition methods: audio, visual, and spontaneous expressions, IEEE Trans. Pattern Anal. Mach. Intell. 31 (1) (2008) 39–58.
[19] G. Zehfuss, Über eine gewisse determinante, Z. für Math. und Phys. 3 (1858) 298–301.
[20] F. Bach, Learning with submodular functions: a convex optimization perspective, Found. Trends Mach. Learn. 6 (2011) 2.
[22] S.S. Bucak, R. Jin, A.K. Jain, Multi-label learning with incomplete class assignments, in: CVPR, IEEE, Colorado Springs, CO, USA, 2011, pp. 2801–2808.
[23] R.S. Cabral, F.D.L. Torre, J.P. Costeira, A. Bernardino, Matrix completion for multi-label image classification, in: NIPS, 2011, pp. 190–198.
[24] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, The extended cohn-kanade dataset (ck+): a complete dataset for action unit and emotion-specified expression, in: CVPR Workshop, 2010, pp. 94–101.
[25] G. Mckeown, M.F. Valstar, R. Cowie, M. Pantic, M. Schroeder, The semaine database: annotated multimodal records of emotionally coloured conversations between a person and a limited agent, IEEE Trans. Affect. Comput. 3 (1) (2013) 5–17.
[26] Y.L. Tian, T. Kanade, J.F. Cohn, Recognizing action units for facial expression analysis, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 97–115.
[27] M. Valstar, M. Pantic, Fully automatic recognition of the temporal phases of facial actions, IEEE Trans. Syst., Man, Cybern., B Cybern. 42 (1) (2012) 28–43.
[28] Y. Chang, C. Hu, M. Turk, Probabilistic expression analysis on manifolds, in: CVPR, IEEE, Washington, DC, USA, 2004, pp. 520–527.
[29] I. Cohen, N. Sebe, F. Cozman, M. Cirelo, T. Huang, Learning Bayesian network classifiers for facial expression recognition both labeled and unlabeled data, in: CVPR, IEEE, 2003, pp. 595–601.
[30] M.S. Bartlett, G. Littlewort, M.G. Frank, C. Lainscsek, I. Fasel, J.R. Movellan, Recognizing facial expression: machine learning and application to spontaneous behavior, in: CVPR, IEEE, San Diego, CA, USA, 2005, pp. 568–573.
[31] J. Whitehill, C.W. Omlin, Haar features for FACS AU recognition, in: FG, IEEE, Southampton, UK, 2006, pp. 217–222.
[32] Y. Wang, H. Ai, B. Wu, C. Huang, Real time facial expression recognition with AdaBoost, in: ICPR, 2004, pp. 926–929.
[33] C. Shan, S. Gong, P.W. McOwan, Facial expression recognition based on local binary patterns: a comprehensive study, Image Vision. Comput. 27 (6) (2009) 803–816.
[34] Y. Guo, G. Zhao, M. Pietikäinen, Discriminative features for texture description, Pattern Recognit. 45 (10) (2012) 3834–3843.
[35] Y. Zhu, L.T.F. De, J.F. Cohn, Y.J. Zhang, Dynamic cascades with bidirectional bootstrapping for action unit detection in spontaneous facial behavior, IEEE Trans. Affect. Comput. 2 (2) (2011) 79–91.
[37] Y. Tong, J. Chen, Q. Ji, A unified probabilistic framework for spontaneous facial activity modeling and understanding, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2) (2010) 258–273.
[38] Y. Zhu, S. Wang, L. Yue, Q. Ji, Multiple-facial action unit recognition by shared feature learning and semantic relation modeling, in: ICPR, IEEE, Stockholm, Sweden, 2014, pp: 1663–1668.
[39] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, R. Stat. Soc. Ser. B 39 (1) (1977) 1–38.
[40] S. Geman, D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 721–741.
[41] W. Liao, Q. Ji, Learning Bayesian network parameters under incomplete data with domain knowledge, Pattern Recognit. 42 (11) (2009) 3046–3056.
[42] S.S. Bucak, R. Jin, A.K. Jain, Multi-label learning with incomplete class assignments, in: CVPR, IEEE, Colorado Springs, CO, USA, 2011, pp. 2801–2808.
[43] Q. Wang, L. Si, D. Zhang, Learning to hash with partial tags: exploring correlation between tags and hashing bits for large scale image retrieval, in: ECCV, 2014, pp. 378–392.
[44] Y.Y. Sun, Y. Zhang, Z.H. Zhou, Multi-label learning with weak label, in: AAAI, 2010.
[45] R.S. Cabral, F. De la Torre, J.P. Costeira, A. Bernardino, Matrix completion for multi-label image classification, in: NIPS, 2011, pp. 190–198.

[46] A. Goldberg, B. Recht, J.M. Xu, R. Nowak, X. Zhu, Transduction with matrix completion: three birds with one stone, in: NIPS, 2010, pp. 757–765.
[47] M. Xu, R. Jin, Z.H. Zhou, Speedup matrix completion with side information: application to multi-label learning, in: NIPS, 2013, pp: 2301–2309.
[48] B. Jiang, M. Valstar, M. Pantic, Action unit detection using sparse appearance descriptors in space-time video volumes, in: FG Workshops, IEEE, Santa Barbara, CA, USA, 2011, pp. 314–321.
[49] Z. Wang, Y. Li, S. Wang, Q. Ji, Capturing global semantic relationships for facial action unit recognition, in: ICCV, IEEE, Sydney, Australia, 2013, pp. 3304-3311.
[50] X. Zhang, M.H. Mahoor, Simultaneous detection of multiple facial action units via hierarchical task structure learning, in: ICPR, IEEE, Stockholm, Sweden, 2014, pp. 1863–1868.
[51] X. Zhang, M.H. Mahoor, S.M. Mavadati, J.F. Cohn, An lp-norm MTMKL framework for simultaneous detection of multiple facial action units, in: WACV, IEEE, Steamboat Springs, CO, USA, 2014, pp. 1104–1111.
[52] S. Eleftheriadis, O. Rudovic, M. Pantic, Multi-conditional latent variable model for joint facial action unit detection, in: ICCV, IEEE, Santiago, Chile, 2015, pp. 3792–3800.
[53] D. Cai, X. He, J. Han, SRDA: an efficient algorithm for large-scale discriminant analysis, IEEE Trans. Knowl. Data Eng. 20 (1) (2008) 1–12.
[54] Y. Zhang, Z.H. Zhou, Multilabel dimensionality reduction via dependence maximization, ACM Trans. Knowl. Discov. Data 4 (3) (2008) 1503–1505.
[55] J.R. Shewchuk, Technical report: an introduction to the conjugate gradient method without the agonizing pain, Carnegie Mellon University, Pittsburgh, PA, USA, 2010.
[56] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1124–1137.
[57] X. Zhu, Semi-supervised learning literature survey, Citeseer (2005).
[58] U.V. Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[59] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, UK, 2004.
[60] B. Wu, S. Lyu B. Ghanem, Constrained submodular minimization for missing labels and class imbalance in multi-label learning, in: AAAI, 2016.

Yongqiang Li received the BS, MS and PhD degrees in instrument science and technology from Harbin Institute of Technology, Harbin, China, in 2007, 2009 and 2014, respectively. He is currently an assistant professor at Harbin Institute of Technology. He worked as a visiting student at Rensselaer Polytechnic Institute, Troy, USA, from September 2010 to September 2012. His areas of research include computer vision, patter recognition, and human–computer interaction.

Baoyuan Wu received his PhD degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy in 2014. He was a visiting student in Rensselaer Polytechnic Institute, from September 2011 to September 2013. He is currently a Postdoc in King Abdullah University of Science and Technology (KAUST). His main research interests include probabilistic graphical models, multi-label learning, clustering and integer programming.

Bernard Ghanem is currently an assistant professor in the CEMSE division and a member of the Visual Computing Center at KAUST. Before that, he was a senior research scientist at the University of Illinois Urbana-Champaign (UIUC) in Singapore, where he still holds an adjunct position. He heads projects that develop algorithms in computer vision, machine learning, and optimization geared towards real-world applications, including semantic video analysis in sports and automated surveillance, content-based image retrieval, large-scale activity recognition, and 2D/3D scene understanding. He received his Bachelor's degree in Computer and Communications Engineering from the American University of Beirut (AUB) in 2005 and his MS/PhD in Electrical and Computer Engineering from UIUC in 2010. His work has received several awards and honors, including the Henderson Graduate Award from UIUC, two consecutive CSE fellowship awards from UIUC, a Best Paper Award (CVPRW 2013), a two-year KAUST Seed Fund, and a Google Faculty Research Award in 2015. He has co-authored more than 40 peer reviewed conference and journal papers in his field, as well as, 4 patents. He is also a co-founder of AutoScount Inc. that provides automated solutions for sports video analytics.

Yongping Zhao received the PhD degree in electrical engineering from Harbin Institute of Technology, Harbin, China. He is currently a professor with the Department of Instrument Science and Technology at Harbin Institute of Technology, Harbin, China. His areas of research include signal processing, system integration and patter recognition.

Hongxun Yao received the BS and MS degrees in computer science from the Harbin Shipbuilding Engineering Institute, Harbin, China, in 1987 and in 1990, respectively, and the PhD degree in computer science from Harbin Institute of Technology in 2003. Currently, she is a professor with the School of Computer Science and Technology, Harbin Institute of Technology. Her research interests include pattern recognition, multimedia technology, and human–computer interaction technology. She has published three books and over 100 scientific papers.

Qiang Ji received his PhD degree in Electrical Engineering from the University of Washington. He is currently a professor with the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute (RPI). He recently served as a program director at the National Science Foundation (NSF), where he managed NSF's computer vision and machine learning programs. He also held teaching and research positions with the Beckman Institute at University of Illinois at Urbana-Champaign, the Robotics Institute at Carnegie Mellon University, the Department of Computer Science at University of Nevada at Reno, and the US Air Force Research Laboratory. Prof. Ji currently serves as the director of the Intelligent Systems Laboratory (ISL) at RPI.

Prof. Ji's research interests are in computer vision, probabilistic graphical models, information fusion, and their applications in various fields. He has published over 160 papers in peer-reviewed journals and conferences. His research has been supported by major governmental agencies including NSF, NIH, DARPA, ONR, ARO, and AFOSR as well as by major companies including Honda and Boeing. Prof. Ji is an editor on several related IEEE and international journals and he has served as a general chair, program chair, technical area chair, and program committee member in numerous international conferences/workshops. Prof. Ji is a fellow of IAPR and a fellow of the IEEE.