Action Unit Intensity Estimation using Hierarchical Partial Least Squares

Tobias Gehrig[1,*], Ziad Al-Halah[1,*], Hazım Kemal Ekenel[2], Rainer Stiefelhagen[1]
[1] Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany
[2] Faculty of Computer and Informatics, Istanbul Technical University, Istanbul, Turkey

***Abstract*— Estimation of action unit (AU) intensities is considered a challenging problem. AUs exhibit high variations among the subjects due to the differences in facial plasticity and morphology. In this paper, we propose a novel framework to model the individual AUs using a hierarchical regression model. Our approach can be seen as a combination of locally linear Partial Least Squares (PLS) models where each one of them learns the relation between visual features and the AU intensity labels at different levels of details. It automatically adapts to the non-linearity in the source domain by adjusting the learned hierarchical structure. We evaluate our approach on the benchmark of the Bosphorus dataset and show that the proposed approach outperforms both the 2D state-of-the-art and the plain PLS baseline models. The generalization to other datasets is evaluated on the extended Cohn-Kanade dataset (CK+), where our hierarchical model outperforms linear and Gaussian kernel PLS.**

I. INTRODUCTION

Analysis of facial expressions is becoming more important, e.g. in order to adapt to the current state of the person interacting with a computer-interface or in psychological analysis [5]. For most applications, facial expression analysis is reduced to the recognition of prototypic facial expressions from emotions like happy, sad, angry, disgust, surprise, or fear. While this simplifies the task of classification, it does not provide recognition of the whole spectrum of emotional facial expressions. Moreover, the prototypical expressions rarely occur in their idealized form in real-life. Also, treating these prototypic expressions equivalent to their emotional origins is dangerous, since the expression can also originate from a different affective state. For example, a smiling face might originate from happiness, but also from frustration [2]. Therefore, in this work we concentrate on analyzing the facial expression without interpreting its meaning in order to lay the ground for later interpretation. The most commonly used description scheme in this context is the *Facial Action Coding System* (FACS) [3], which describes the facial expression in terms of *Action Units* (AU) on basis of the visual effect of facial muscle activations. There are a lot of studies proposing methods for automatically detecting the presence of such AUs [23]. However, AUs don't appear in a natural scenario just with high activation or no at all, rather with continuous intensity variations. Our focus in this work lies on estimating the intensity of AUs in a continuous fashion, even though the FACS manual [3] discretizes the intensities into six classes, 0 and A to E, to ease human annotation. This would also allow to analyze subtle expressions which might not be detected if only the presence of AUs would be classified.

Fig. 1: Examples of face images that show variations in performing the same AUs, either as a result of differences in face morphology (first line) or recording settings (second line). (Best viewed in color)

AU intensity estimation is a field that has not received as much attention as other fields of facial expression analysis. Pantic and Rosenkrantz [12] developed an expert system to recognize AU intensities on basis of detected facial features. Bartlett et al. [1] showed that the distance to the hyperplane of Support Vector Machines (SVM) trained for AU detection is correlated with the actual AU intensity, when using Gabor Wavelets as features. Most of the approaches use variants of SVMs, like multi-class SVM [10], [21], [22], [11] or Support Vector Regression (SVR) [17], [18], [7]. Other machine learning approaches used are Relevant Vector Regression [8], Gaussian Process Regression [6], Markov Random Fields [15], Conditional Ordinal Random Fields [14].

In general, such systems first extract some suitable face representation, which is then used as input to some machine learning approach for estimating the AU intensities. The relation between these low-level features and the AU intensities is usually non-linear. This non-linearity comes from different factors; some are related to the extracted visual features and their robustness to changes in lighting, scale and perspective among others, while others are related to the dynamics of the AUs relative to subjects. Different people exhibit significant differences in performing the various AUs.

* Both authors contributed equally to this study.

This work was partially funded by BMBF and TUBITAK under the Intensified Cooperation Programme (IntenC), project no. 01DL13016 and 113E067, and a Marie Curie FP7 Integration Grant within the 7th EU Framework Programme.

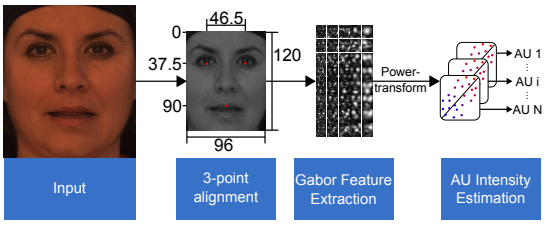


Fig. 2: Flow diagram of the feature extraction and AU intensity estimation of our approach.

These differences are related to the varying face properties which are caused by differences in gender (*e.g.* changes in eyebrows), age (*e.g.* changes in skin and muscle structure), and race (*e.g.* changes in shape) [20], as can be seen in Fig. 1.

A prominent approach to model such non-linearity is by using kernel models [18]. However, kernel models generally require a notably higher computational cost in both time and memory while at the same time tend to generalize poorly for out of sample estimation. In this work, we propose to model the non-linear relation with a set of linear sub-models. Each of these takes advantage of the local linearity of a subset of the problem space. This enables us to have an AU model that is robust to the previous factors, *i.e.* subject and recording settings variations, while at the same time maintain an efficient and effective learning.

In this context, we introduce a novel hierarchical model based on *Partial Least Squares* (PLS) [13]. Our model can handle non-linearities in the data by utilizing combinations of locally linear PLS sub-models, each of which is learned at different levels of detail. Such sub-models are computationally cheap to compute while at the same time they enable the overall model to capture non-linearity in an efficient manner compared to their kernel counterparts. The structure of the model is learned automatically to adapt to the type of the relation between each AU and the visual features.

II. Approach

The flow diagram for the utilized feature extraction and AU intensity estimation of our approach is shown in Fig. 2. In the following, we discuss each module with more details.

A. Feature Extraction

The feature extraction starts with aligning the face based on the eyes and the center of the upper inner lip to reduce variations in in-plane rotation, small out-of-plane rotations, scaling, and person-specific face shapes. For the positions we use the labels provided with the databases. The alignment is performed such that the eyes and the upper lip always end up on the same points in the aligned image, specified by the eye row, eye distance and upper lip row. For this an affine transformation is utilized. The resulting aligned face image is then cropped and converted to grayscale for further feature extraction using a Gabor filter bank. To reduce the dimensionality of the Gabor features we downscale the Gabor filtered images and concatenate all the resulting features into one feature vector. A power transform $p(x) = x^\gamma$ is applied to each element of this feature vector to stabilize its variance and make it more like a normal distribution. Finally, the feature vector as well as the intensity labels are normalized to be zero-mean and scaled to unit variance before performing any training or estimation.

B. Partial Least Square Analysis

As a core regression model for AU estimation, we use *partial least squares* as described by Rosipal [13]. This is a general concept, which has been used a lot in the field of chemometrics and also in the recent years in computer vision. It relates input and output variables $\mathbf{X}$ and $\mathbf{Y}$ via an intermediate latent space. This space is trained to maximize the covariance between projections t and u from input and output space into the latent space:

$$[cov(\mathbf{t}, \mathbf{u})]^2 = \max_{|\mathbf{r}|=|\mathbf{s}|=1} [cov(\mathbf{Xr}, \mathbf{Ys})]^2 \quad (1)$$

This leads to a compact representation in the latent space and higher weights for task relevant features via the supervised PLS training. More details about the characteristics of the PLS model can be found in [13].

To calculate the projection $\mathbf{T}$ from the input data $\mathbf{X}$ to the latent space, we use the projection matrix $\mathbf{W}$ learned as part of the PLS model:

$$\mathbf{T} = \mathbf{XW} \quad (2)$$

The corresponding AU intensity $f_{PLS}(\mathbf{X})$ is estimated using the learned regression matrix $\mathbf{B}$:

$$f_{PLS}(\mathbf{X}) = \mathbf{XB} \quad (3)$$

In this work, we use the kernel PLS as a baseline and compare results for linear and Gaussian *radial basis function* (RBF) kernels:

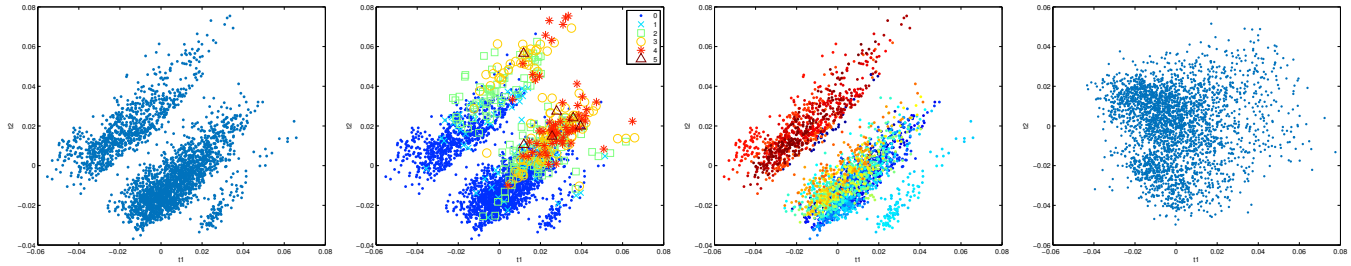
$$k_{\text{linear}}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j \quad (4)$$

$$k_{\text{gaussian}}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-(|\mathbf{x}_i - \mathbf{x}_j|^2)/w), \quad (5)$$

C. Hierarchical Partial Least Squares

In the following, we present the proposed hierarchical approach and its various steps: i) the identification and learning of locally linear models; ii) model selection and iii) model combination.

Model Hierarchy A key factor for the performance of locally linear models is the methodology adopted to capture and identify local linear subsets of the problem space. To identify these subsets in the input-output space, we propose to take advantage of the latent space learned by PLS as a cue. As described in Section II-B, PLS maximizes the covariance between the projections of the features and AU estimations into the latent space. The non-linear relation between the two spaces, features and AU intensities, is usually reflected in the properties of the learned latent space. Fig. 3 shows the latent space learned by PLS for AU4 (Brow Lowerer). One can notice the existence of two distinct groups. The clusters are not caused by changes in AU intensities, rather they follow a clear split among the subjects. In this case, the split is due to a sudden change in the lighting conditions of



(a) AU4 latent space (b) AU4 intensities (c) AU4 subjects (d) AU12 latent space

Fig. 3: The PLS learned latent space of action unit AU4 represented with the first two latent components (a). Figure (b) labels the latent space with AU intensities while in (c) the subject labels are shown. Notice that the split between the samples in the latent space follows a division among the subjects not the intensities. However, the same grouping does not appear in all AUs as seen in the latent space of AU12 in (d). Best viewed in color.

the recorded samples. On the other hand, not all AUs seem to follow a similar behavior. For example, while the lighting changes have a high impact on AU4 it doesn't have a similar influence on AU12 (Lip Corner Puller), as shown in Fig. 3d.

Based on the previous observation, we propose to decompose the input-output space following the grouping exposed by the learned latent space of PLS in an iterative manner. That is, first a global linear PLS model for a certain AU is learned using all samples available. Then, samples are projected to the latent space using (2). Latent scores are clustered into S groups. If the clustering is good, the result is mapped back to input (visual features) and output space (AU intensities). For each subset $\{(\mathbf{X}_i, \mathbf{Y}_i) : i = 1...S\}$, a new PLS model is learned as in Section II-B. The procedure is repeated as long as a good split in the latent space can be detected (Algo. 1).

At the end of this step, a hierarchical PLS model (hPLS) is learned. Where each node in the hierarchy is a locally linear PLS model that captures the AU-feature relation at a certain abstraction level. The root of the tree models the global relation between $\mathbf{X}$ and $\mathbf{Y}$ while the subsequent nodes model the finer details among the various subsets in the dataset. Notice that the approach iteratively splits and learns local models. Hence, the number of clusters S does not play an important role and can be set to an arbitrarily low value (*e.g.* $S = 2$, *i.e.* binary tree). That is, if a larger number of clusters appears in the latent space, the model will probably discover this in a later stage in the hierarchy.

Model Selection In order to evaluate the "goodness" of a certain split, we need some kind of a quality measurement. We adopt the corrected Akaike Information Criterion (AICc). AICc measures the estimated relative loss of information when a model is selected to describe the system that produced the data:

$$\mathrm{AICc} = 2(k - \ln(L)) + \tfrac{2k(k+1)}{n-k-1}, \tag{6}$$

where k is the number of model parameters, n is the number of samples and L is the maximized likelihood value. Hence, to assess the quality of a certain split we check the relative loss exhibited when adopting the split into S groups compared to no split, *i.e.* :

$$\exp((\mathrm{AICc}_0 - \mathrm{AICc}_S)/2) > \theta. \tag{7}$$

input : $\mathbf{X}$, $\mathbf{Y}$, (S=2)
output: model
$PLS_{root} \leftarrow \mathrm{PLS\_fit}(\mathbf{X}, \mathbf{Y})$
$(model, root) \leftarrow \mathrm{add\_to\_hierarchy}([\,], PLS_{root})$
$node\_list \leftarrow \mathrm{push}(root, node\_list)$
while *not is_empty(node_list)* **do**
  $n_i \leftarrow \mathrm{pop}(node\_list)$
  $lt_{n_i} \leftarrow \mathrm{get\_latent\_space}(\mathbf{X}_{n_i}, \mathbf{Y}_{n_i}, PLS_{n_i})$
  $C_{n_i} \leftarrow \mathrm{cluster}(lt_{n_i}, S)$
  if *not is_good*(C_{n_i}) **then**
    continue
  end
  for $j \leftarrow 1$ **to** S **do**
    $idx_j \leftarrow \mathrm{get\_cluster}(j, C_{n_i})$
    $PLS_j \leftarrow \mathrm{PLS\_fit}(\mathbf{X}_{n_i}(idx_j), \mathbf{Y}_{n_i}(idx_j))$
    $(model, n_j) \leftarrow \mathrm{add\_to\_hieararchy}(n_i, PLS_j)$
    $node\_list \leftarrow \mathrm{push}(n_j, node\_list)$
  end
end

Algorithm 1: Hierarchical PLS.

Model Combination Once the hierarchical model is learned, AU intensity estimations can be retrieved by combining the models at different levels of the hierarchy. That is, a new sample is projected into the latent space of the root node in the tree and classified down to a leaf node n_l. All PLS models along the tree branch of the sample are combined to get the average estimation of the AU:

$$f_{\mathrm{hPLS}}(x) = \tfrac{1}{|\mathrm{anc}(n_l)|} \sum_{n_j \in \mathrm{anc}(n_l)} f^{n_j}_{\mathrm{PLS}}(x) \quad \text{such that } x \in n_l, \tag{8}$$

where $\mathrm{anc}(n)$ is the set of ancestors of node n.

By combining the models through the hierarchy, we take advantage of the characteristics of the models at each level. The models towards the root are trained with larger subsets. Hence, they provide a more stable (less sensitive to noise) and general estimation of the AU intensities. On the other hand, the models towards the leaves are trained with smaller subsets and they provide a less stable but more accurate estimation of the AU intensities.

Our approach has some nice properties. It automatically adapts to the varying complexity of the relation between the

feature space and the AU intensity estimates for the different AUs. It learns the appropriate structure to model this relation by varying the tree width and depth. It is also a generic model, *i.e.* any other regression model (*e.g.* CCA) can easily replace the PLS in the hierarchy to have a novel model with different properties.

III. EXPERIMENTS

To have a better understanding of the generalization properties of the various methods, we evaluate our approach using two setups: within-dataset intensity estimation, *i.e.* training and testing in the same dataset; and across-dataset intensity estimation, *i.e.* training and testing conducted on disjoint datasets. Furthermore, we will discuss the computational efficiency of the proposed approach.

A. Datasets

For the within-dataset experiment, we chose the Bosphorus dataset [16]. It contains 2902 images of 105 subjects with a rich variety of acted expressions. The samples are labeled with FACS AU intensities. The labels were provided by a certified FACS coder. The subjects are mostly Caucasian between 25 and 35 years with varying facial characteristics.

For the across-dataset experiment, we picked the Extended Cohn-Kanade (CK+) dataset [9]. It contains 117 frames of 73 subjects labeled with 24 AU intensities. As Bosphorus, this dataset is coded by a certified FACS coder.

B. Evaluation Setup

The evaluation is carried out following the official benchmark proposed by Savran et al. [18]. 25 AUs in the dataset with all five intensity levels, *i.e.* excluding 0 intensity, are evaluated separately where each AU has a predefined 10-fold cross validation setup. Each fold is chosen such that the subjects do not occur in multiple folds and the folds are balanced in terms of the number of positive AU samples.

The performance in the official Bosphorus benchmark is measured using the *Pearson correlation coefficient* (PCC) between the estimated and the ground-truth AU intensities. In addition to PCC, we use the *Intraclass Correlation Coefficient* (ICC) [19] for determining the performance, since it is said to be preferred over PCC when it comes to computing the consistency between k judges [10]. The reason for this is that labels and estimates are centered and scaled by a common mean and standard deviation rather than individually as for PCC. Finally, the weighted average of the PCC and ICC results over the individual AUs are used as the overall performances, where the weights are determined by the number of AU samples.

For the feature extraction, we align eyes to be on row 37.5, to have an interocular distance of 46.5 pixels, and the inner upper lip center to lie on row 90. The Gabor features are downscaled by a factor of 8, *i.e.* for each block of 8×8 we replace it by its average value. For the power transformation we use $\gamma = 0.25$. The parameters for the various PLS models are estimated using 5-fold cross-validation on the corresponding training folds. We use one PLS model per AU due to the way the folds in the official benchmark for the Bosphorus database are defined. But PLS can also estimate all AUs at once [4].

TABLE I: Average performance of AU intensity estimation in terms of PCC and ICC evaluated using 10-fold cross-validation on the official benchmark folds of Bosphorus.

Metric	hPLS (ours)	linear-PLS	RBF-PLS	AdaSVR [18]
PCC	**61.7**	59.6	60.5	57.6
ICC	**57.9**	57.3	56.0	-

C. Within-dataset AU intensity estimation

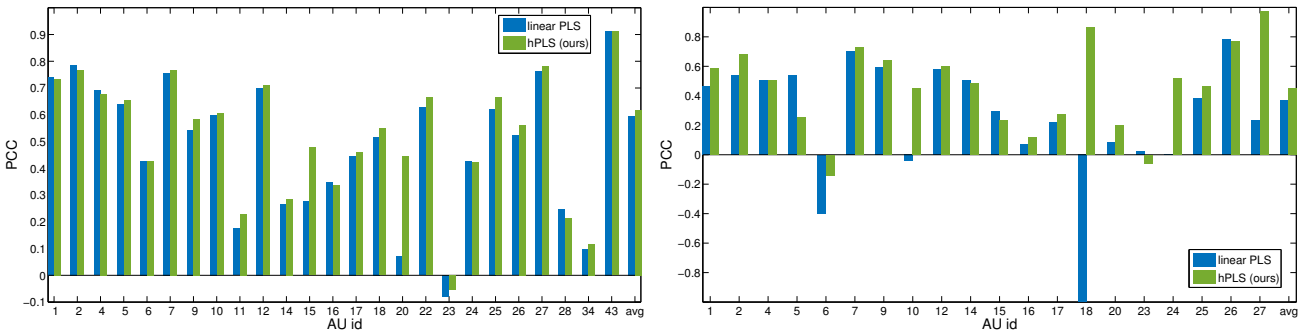
In the first evaluation, we compare the plain PLS baseline and our proposed hierarchical PLS approach with the state-of-the-art 2D method from Savran et al. [18]. Their approach uses Gabor magnitude features extracted from images aligned based on the eyes position. From these features a subset is selected by AdaBoost. The AU intensity estimation is performed by an RBF *support vector regression* (ϵ-SVR). From here on, we will refer to this approach as AdaSVR.

The weighted average PCC and ICC results over all AUs for the hierarchical PLS, linear PLS, RBF kernel PLS, and AdaSVR are shown in Table I. We notice that all our results outperform AdaSVR. In initial experiments, we also evaluated 2-point alignment as used by AdaSVR and found that, while the plain PLS with linear and RBF kernels performed as well as AdaSVR, our model (hPLS) achieves 59.7% PCC outperforming all. In general, the 3-point alignment enhanced the performance of the plain PLS baseline and our hPLS. This is expected, since due to different face shapes the mouth will be differently positioned for different persons in case of sole eye-based alignment, whereas for 3-point alignment the mouth will be positioned at roughly the same region for different persons.

Additionally, the use of the RBF kernel improves over the linear PLS by 0.9% absolute difference in PCC, but drops in performance in terms of ICC by 1.3%. On the other hand, RBF PLS takes much longer time to train due to the more complex parameter estimation, whereas the hierarchical PLS is much faster in training due to its locally linear concept. hPLS improves over linear and RBF PLS in both PCC and ICC. This shows that the hierarchical approach seems to actually model the non-linearity even better than the RBF PLS does, while keeping the complexity of training low.

When we look at the individual performances for the various AUs in Fig. 5a, we notice that there are 18 AUs where the hierarchical PLS performs better compared to just 8 AUs where the linear PLS works slightly better. For AUs, such as 15 (Lip Corner Depressor) and 20 (Lip Stretch), the hierarchical PLS gives a significant performance boost.

To get some insight in the structure of the proposed hPLS, Fig. 4 shows the average number of locally linear models in the learned hierarchy for each AU, while in Fig. 6 some of the splits learned by hPLS at different levels and for various AUs are presented.



(a) Bosphorus (b) CK+

Fig. 5: The performance of the individual AUs in terms of PCC compared to the linear PLS model in both (a) Bosphorus and (b) CK+ datasets.

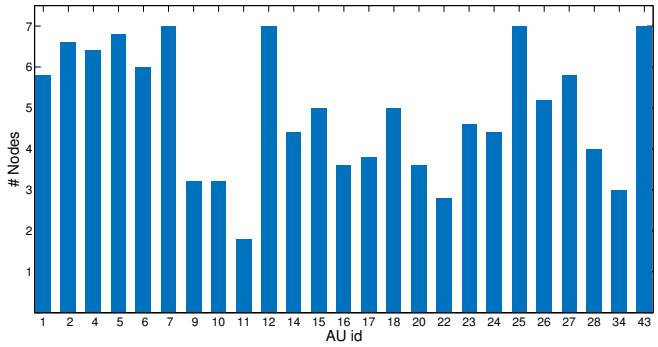


Fig. 4: The average number of nodes in the learned hierarchy for each AU in the Bosphorus dataset. Our model automatically adapts to the complexity of each AU by varying the number of learned locally linear submodels.

TABLE II: Average performance of AU intensity estimation in terms of PCC and ICC for across-dataset evaluation, *i.e.* training on Bosphorus and testing on CK+.

Metric	hPLS (ours)	linear-PLS	RBF-PLS
PCC	**44.9**	37.0	41.2
ICC	**41.2**	35.8	26.2

D. Across-dataset AU intensity estimation

To demonstrate the generalization ability of our approach across datasets, we evaluate its performance when trained on Bosphorus and tested on CK+, since the latter has too few samples labeled with AU intensities to use for training. Like for the official Bosphorus benchmark, we only consider samples with intensity > 0 for both training and testing.

The weighted average PCC and ICC results over all AUs for the hierarchical PLS, linear PLS, and RBF kernel PLS are shown in Table II. As expected, the hierarchical PLS gives a much higher generalization performance than the plain PLS approaches in both metrics. Moreover, we notice that the RBF PLS shows inconsistent performance with a significant drop in terms of ICC compared to the linear PLS. This supports the claim that using a linear rather than a non-linear kernel has better generalization properties across-datasets.

When we look at the individual performances for the various AUs in Fig. 5b, we notice that compared to the within-dataset experiment on Bosphorus there are more AUs that benefit from the hPLS model, such as 1 (Inner Brow Raise), 2 (Outer Brow Raise), 10 (Upper Lip Raiser), 20 (Lip Stretch), 24 (Lip Presser), and 27 (Mouth Stretch), for which the hierarchical PLS gives a significant performance boost. This shows that for most of the AUs the hierarchical PLS generalizes better to new out-of-domain data.

E. Computional Efficiency

As we mentioned earlier, the hPLS has lower computational costs than the RBF kernel PLS. We investigate this statement in more details in this subsection based on the runtime of the crossvalidation experiment on Bosphorus. When looking at the complexity alone, one can see that for RBF PLS, the kernel matrix calculation is much slower than using simply the data matrix like in linear PLS, and also the rest of the optimization in kernel PLS is more computationally demanding. Additionally, the parameter estimation in case of the hPLS needs to search just for the number of latent variables for each node, which is much faster for linear PLS than for non-linear PLS, since no kernel matrix needs to be calculated. The average number of locally linear PLS in the learned hierarchy for the individual AUs is shown in Fig. 4. On the other hand, for RBF kernel PLS the search for parameters has to go over w in addition to the number of latent variables. The runtime of the crossvalidation over all AUs on Bosphorus including I/O and parameter estimation took around 32 minutes for a Matlab implementation of hPLS, whereas the runtime for the RBF kernel PLS with a slightly larger search space for our C++ implementation is more than 7 hours.

IV. CONCLUSIONS AND FUTURE WORK

We have presented a hierarchical regression model for AU intensity estimation. Our model can automatically capture the non-linearity in the relation between the features and the intensities of the individual AUs. It adapts to the varying complexity in the domain by learning a suitable hierarchical structure. The proposed approach outperformed the more complex kernel-based models while at the same time having much lower computational costs.

As a future direction, we intend to extend our model to incorporate some prior knowledge of the problem domain.

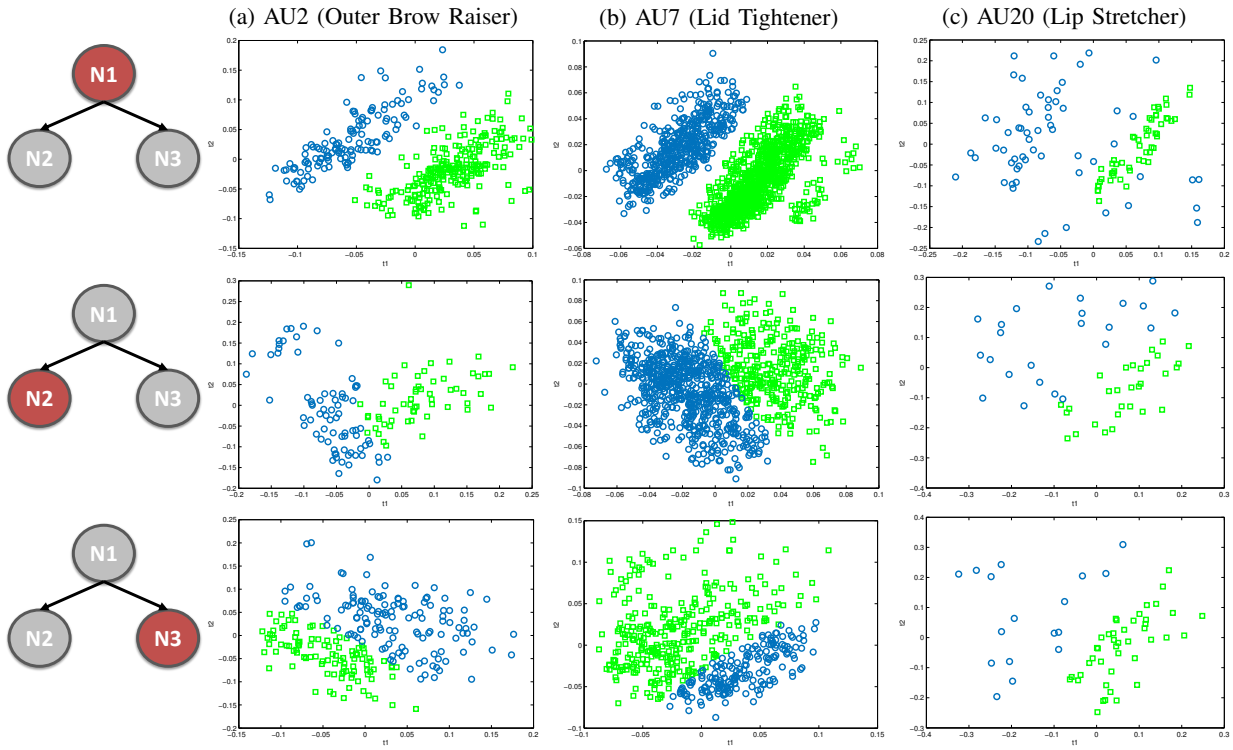


Fig. 6: The latent space learned by the local PLS models in the first two levels of the hierarchy and for different AUs.

In this paper, the model learned the structure driven entirely by the data. Incorporating knowledge about the participant subjects can be promising. For example, by using visual facial attributes or gender and race information to learn a better grouping of the samples in the local subsets the prior information could guide the clustering towards grouping more visually similar persons.

REFERENCES

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

[2] M. (Ehsan) Hoque and R. W. Picard. Acted vs. natural frustration and delight: Many people smile in natural frustration. In *FG*, 2011.

[3] P. Ekman, W. V. Friesen, and J. C. Hager. *Facial Action Coding System - The Manual*. 2002.

[4] T. Gehrig and H. K. Ekenel. Facial Action Unit Detection Using Kernel Partial Least Squares. In *1st IEEE Int'l Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[5] J. M. Girard, J. F. Cohn, Mohammad H. Mahoor, S. Mavadati, and D. P. Rosenwald. Social risk and depression: Evidence from manual and automatic facial expression analysis. In *FG*, 2013.

[6] D. Haase, M. Kemmler, O. Guntinas-Lichius, and J. Denzler. Efficient Measuring of Facial Action Unit Activation Intensities using Active Appearance Models. In *IAPR International Conference on Machine Vision Applications*, 2013.

[7] L. A. Jeni, J. M. Girard, J. F. Cohn, and F. De la Torre. Continuous AU intensity estimation using localized, sparse facial feature space. In *FG*, 2013.

[8] S. Kaltwang, O. Rudovic, and M. Pantic. Continuous Pain Intensity Estimation from Facial Expressions. *Advances in Visual Computing*, 2012.

[9] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, and Z. Ambadar. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *The 3rd IEEE Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB)*, 2010.

[10] M. H. Mahoor, S. Cadavid, D. S. Messinger, and J. F. Cohn. A framework for automated measurement of the intensity of non-posed Facial Action Units. In *CVPR Workshops*, 2009.

[11] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn. DISFA: A Spontaneous Facial Action Intensity Database. *IEEE Transactions on Affective Computing*, 2013.

[12] M. Pantic and L. J.M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE T-PAMI*, 2000.

[13] R. Rosipal. Nonlinear Partial Least Squares: An Overview. In *Chemoinformatics and Advanced Machine Learning Perspectives: Complex Computational Methods and Collaborative Techniques*. ACCM, IGI Global, 2011.

[14] O. Rudovic, V. Pavlovic, and M. Pantic. Context-sensitive Conditional Ordinal Random Fields for Facial Action Intensity Estimation. In *HACI*, 2013.

[15] G. Sandbach, S. Zafeiriou, and M. Pantic. Markov Random Field Structures for Facial Action Unit Intensity Estimation. In *ICCV Workshop*, 2013.

[16] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus Database for 3D Face Analysis. In *The First COST 2101 Workshop on Biometrics and Identity Management*, 2008.

[17] A. Savran, B. Sankur, and M. T. Bilge. Estimation of Facial Action Intensities on 2D and 3D Data. In *19th European Signal Processing Conference*, 2011.

[18] A. Savran, B. Sankur, and M. T. Bilge. Regression-based Intensity Estimation of Facial Action Units. *Image and Vision Computing*, 2012.

[19] P. E. Shrout and J. L. Fleiss. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 1979.

[20] Y.-L. Tian, T. Kanade, and J. F. Cohn. *Facial Expression Analysis*, chapter 11. Springer, 2005.

[21] N. Zaker, M. H. Mahoor, and W. I. Mattson. Intensity Measurement of Spontaneous Facial Actions: Evaluation of Different Image Representations. In *International Conference on Development and Learning*, 2012.

[22] N. Zaker, M. H. Mahoor, W. I. Mattson, D. S. Messinger, and J. F. Cohn. A comparison of alternative classifiers for detecting occurrence and intensity in spontaneous facial expression of infants with their mothers. In *FG*, 2013.

[23] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE T-PAMI*, 2009.