Kernel Conditional Ordinal Random Fields for Temporal Segmentation of Facial Action Units

Ognjen Rudovic[1], Vladimir Pavlovic[2], Maja Pantic[1,3]

[1] Computing Dept., Imperial College London, UK
[2] Dept. of Computer Science, Rutgers University, USA
[3] EEMCS, University of Twente, The Netherlands
{o.rudovic,m.pantic}@imperial.ac.uk,vladimir@cs.rutgers.edu,

Abstract. We consider the problem of automated recognition of temporal segments (neutral, onset, apex and offset) of Facial Action Units. To this end, we propose the Laplacian-regularized Kernel Conditional Ordinal Random Field model. In contrast to standard modeling approaches to recognition of AUs' temporal segments, which treat each segment as an independent class, the proposed model takes into account *ordinal* relations between the segments. The experimental results evidence the effectiveness of such an approach.

Keywords: Action units, histogram intersection kernel, ordinal regression, conditional random field, kernel locality preserving projections.

1 Introduction

The dynamics of facial expressions are crucial for interpretation of observed facial behavior. For example, involuntary initiated (spontaneous) facial expressions are characterized by synchronized, smooth, symmetrical, consistent and reflex-like facial muscle movements whereas voluntary initiated (acted) facial expressions are subject to volitional real-time control and tend to be less smooth, with more variable facial dynamics [1].Facial expression dynamics can be explicitly analyzed by detecting the temporal segments (neutral, onset, apex, offset) of facial muscle actions, i.e., Action Units (AUs), and, in turn, their duration, speed and co-occurrences. Although FACS [2] teaches human coders how to identify temporal segments of specific AUs, the manual coding of these segments is labor intensive. Automating this process would make it easier and widely accessible as a research tool [3].

To date, only few works addressed the problem of automatic recognition of temporal segments of AUs in face videos - [4],[5],[6] (frontal view) and [7] (profile view). [6],[7] used rule-based reasoning and geometric based features to encode temporal segments of AUs. [5] used facial-motion-based features and a combination of gentleboost and HMM, while [4] combined SVM and HMM with geometric-based features, for the target task. Note that all these works approach the recognition of AUs' temporal segments as a four-class classification problem. On the other hand, by noting that the AUs' temporal segments are directly related to the intensity of AUs[1], their representation can be enriched with ordinal labels: neutral = 1 ≺ onset, offset = 2 ≺ apex = 3. These *ordinal* relations – which are ignored by the afore-mentioned works – can then be used to augment the classification of the temporal segments. To this end, we propose the Laplacian-regularized Kernel Conditional Ordinal Random Field (Lap-KCORF) model for temporal segmentation of AUs. This model is a non-linear generalization of Conditional Ordinal Random Field (CORF) [8] and its Laplacian-regularized version (Lap-CORF) [9], recently proposed for facial expression intensity estimation. We also propose Composite Histogram Intersection kernel that automatically discovers the facial regions relevant for the classification of the AUs' temporal segmenats.

The remainder of the paper is organized as follows. We give an overview of Conditional Ordinal Random Field (CORF) in Sec. 2. We then describe the proposed Laplacian Kernel CORF model in Sec. 3, and its adaptation for recognition of AU temporal segments in Sec. 4. Sec. 5 shows the experimental results, and Sec. 6 concludes the paper.

2 Conditional Ordinal Random Field (CORF)

The goal of ordinal regression is to predict the output h that indicates the ordinal score of an item represented by a feature vector$\mathbf{x} \in \mathbb{R}^p$. Formally, we let $h = 1 \prec h = 2 \prec \ldots \prec h = R$, where R is the number of ordinal scores. Recently, a CRF-like model named CORF [8] has been proposed for dynamic ordinal regression. Similar to CRF, CORF models the distribution of a set (sequence) of random variables $\mathbf{h}$, conditioned on inputs $\mathbf{x}$. This distribution, denoted by $P(\mathbf{h}|\mathbf{x})$, has a Gibbs form clamped on the observation $\mathbf{x}$ and is defined as:

$$P(\mathbf{h}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\mathbf{x}; \boldsymbol{\theta})} e^{s(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta})}, \tag{1}$$

where $Z(\mathbf{x}; \boldsymbol{\theta}) = \sum_{\mathbf{h} \in \mathcal{H}} e^{s(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta})}$ is the normalizing partition function ($\mathcal{H}$ is a set of all possible output configurations), and $\boldsymbol{\theta} = \{\mathbf{a}, \mathbf{b}, \sigma, \mathbf{u}\}$ are the parameters of the score function $s(\cdot)$, defined as

$$s(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta}) = \sum_{r \in V} \boldsymbol{\Psi}_r^{(V)}(\mathbf{x}, h_r) + \sum_{e=(r,s) \in E} \mathbf{u} \boldsymbol{\Psi}_e^{(E)}(\mathbf{x}, h_r, h_s), \tag{2}$$

where the contribution of static $\Psi_r^{(V)}(\mathbf{x}, h_r)$ and dynamic $\Psi_e^{(E)}(\mathbf{x}, h_r, h_s)$ features is summed over all *node* ($r \in V$) and *edge* ($e = (r, s) \in E$) cliques in the output graph $G = (V, E)$.

[1] The temporal development of an AU starts with its intensity being zero (neutral), followed by the increase in its intensity (onset) until it reaches a peak (apex), from where it decreases (offset) towards zero intensity (neutral).

In contrast to standard CRF, CORF enforces ordering of $\mathbf{h}$ by using the modeling strategy of static ordinal regression methods [9]. Specifically, the probabilistic ranking likelihood, $P(h = c|f_s(x)) = P(f_s(x) \in [b_{c-1}, b_c))$, defined as

$$P(h = c|f(\mathbf{x})) = \Phi\left(\frac{b_c - f_s(\mathbf{x})}{\sigma}\right) - \Phi\left(\frac{b_{c-1} - f_s(\mathbf{x})}{\sigma}\right), \tag{3}$$

is used to set the node features as $\boldsymbol{\Psi}_r^{(V)}(\mathbf{x}, h_r) = \sum_{c=1}^{R} I(h_r = c) \cdot log(P(h_r = c|f_s(\mathbf{x})))$. Here, $\Phi(\cdot)$ is the standard normal cdf, and σ is the parameter that controls the steepness of the likelihood function. The function $f_s(\mathbf{x}) = a^T x$ projects the inputs $\mathbf{x}$ onto a line divided into R bins, where the binning parameters $\mathbf{b} = [-\infty = b_0, \ldots, b_R = +\infty]^\top$ are defined so that they satisfy the ordering constraints ($b_i < b_{i+1}, \forall i$); thus, enforcing the projected features to be sorted according to their ordinal scores h. The edge features, $\boldsymbol{\Psi}_e^{(E)}(\mathbf{x}, h_r, h_s)$, are set as

$$\Big[I(h_r = k \ \wedge \ h_s = l)\Big]_{R\times R} \otimes f_d(\mathbf{x}_r, \mathbf{x}_s). \tag{4}$$

$I(\cdot)$ is the indicator function that returns 1 (0) if the argument is true (false),$\otimes$ is the Kronecker delta, and $f_d(\mathbf{x}_r, \mathbf{x}_s) = |x_r - x_s|$ is the absolute difference between measurement features at adjoining nodes.

Learning of CORF parameters is typically accomplished by maximizing the conditional data likelihood objective (1). In standard log-linear CRF this results in convex optimization [10], while in CORF the objective is nonlinear and nonconvex [9]. Nevertheless, in both cases it is critical to regularize the conditional data likelihood to improve the model performance and generalization. Like CRF, standard CORF [8] uses linear feature function $f_s(x)$ and L_2 regularizer for a. Lap-CORF [9] extends this model using the graph Laplacian regularization. The graph encodes long-term dependencies between the inputs x and imposes general geometric smoothness on CORF predictions. As a consequence, $f_s(x)$ becomes a linear approximation of the general (graph Laplacian) functional leading to better generalization and less sensitivity to inter-subject variations. Although the Lap-CORF representation is effective in some tasks (e.g., [9]), f_s is still constrained to the linear form. This can limit the model's performance if the mappings from the feature space to the ordinal space are highly complex. To address this, in what follows, we generalize Lap-CORF to the non-linear case that permits the use of implicit feature spaces through Mercer kernels.

3 Laplacian Kernel CORF

In this section, we first describe the Kernel CORF (KCORF) model based on the general theory of functional optimization in RKHS. We then use Kernel Locality Preserving Projections [11] to provide specific kernel regularizer for the KCORF model, which gives rise to the Laplacian-regularized KCORF (Lap-KCORF) model. Finally, we explain the learning and inference in the proposed model.

3.1 Kernel CORF

Consider a regularized loss function of the form

$$\arg\min_{\theta,\alpha,\tau} \sum_{i=1}^{N} - \ln P(h_i|f(x_i), \theta, \alpha, \tau) + \Omega_K(\theta, \alpha, \tau), \tag{5}$$

where $\Omega_K(\theta, \alpha, \tau)$ is the (kernel-inducing) regularizer. To find an optimal functional form of $f^*(\cdot)$, Lafferty et al. [12] proposed the following Representer Theorem for conditional graphical models.

Theorem 1 *Let K be a Mercer kernel on $x\, h_C$ with associated RKHS norm $\|\cdot\|_K$, and let $\Omega_K : R_+ \rightarrow R_+$ be strictly increasing. Then the minimizer f^* of optimization problem (5), if it exists, has the form*

$$f^*(\cdot) = \sum_{i=1}^{N} \sum_{c \in C^{(i)}} \sum_{h_c \in h^{|c|}} \alpha_c^{(i)}(h_c) K_c(\mathbf{x}^{(i)}, h_c^{(i)};\ \cdot). \tag{6}$$

Here, c are the vertices of cliques $C^{(i)}$ in graph G, and $h_c \in h^{|c|}$ are all possible labellings of that clique. From (6), we see that the structure in the model output is captured by the 'dual parameters' $\alpha_c^{(i)}$, which depend on all assignments of labels over the training examples. Note, however, that such a model may have an extremely large number of parameters [12]. On the other hand, CORF already models the structure in the output by means of the parametric ordinal model. It also models temporal dynamics of $\mathbf{h}$ by means of the transition matrix $\mathbf{u}$. Therefore, the functional form in (6) can be simplified by dropping dependences on labels $\mathbf{h}$, and by defining kernel K_c only on the node cliques ($r \in V$). In this way, we recover the Kernel CORF model that is the semi-parametric CORF model with static (f_s) and dynamic (f_d) feature functions defined as

$$f_s(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i|\tau) \ \wedge \ f_d(x_t, x_{t-1}) = f_s(x_t) - f_s(x_{t-1}), \tag{7}$$

where τ are the kernel parameters. Note that the above-defined dynamic feature function is defined in the ordinal space, while in the case of CORF and Lap-CORF, it is defined in the ambient space. Thus, the former uses only the relevant features, i.e., projections on the ordinal line that correlate with the segments' intensity, to quantify the temporal changes. By contrast, CORF and Lap-CORF use all (relevant and irrelevant) features to determine this change.

3.2 Regularization using Kernel Locality Preserving Projection

KLPP is the kernel extension of Linear Locality Preserving Projection (LLPP), an optimal linear approximation to the eigenfunctions of the Laplace Beltrami operator on the manifold, that is frequently used for nonlinear dimensionality reduction [11]. KLPP uses the notion of Laplacian of the graph to learn nonlinear mappings that project inputs onto a low-dimensional manifold. Formally, it first constructs an undirected graph $G = (V, E)$ [2], where each edge is associated with a weight W_{ij} that quantifies the similarity of data points (i, j). Given W, KLPP seeks to find the nonlinear function $f(\cdot)$ that is smooth on G, i.e., which minimizes

$$\Omega(\|f\|_K) = \sum_{i=1}^{N} \sum_{j=1}^{N} (f(x_i) - f(x_j))^2 W_{ij} = 2\alpha^T KLK\alpha, \tag{8}$$

where L is graph Laplacian defined as $L = D - W$, and D is a diagonal degree matrix with $D_{ii} = \sum_j W_{ij}$. We derive the similarity measure based on the ordinal labels h as

$$W_{ij} = 1 - \frac{|h_i - h_j|}{R-1},\ h_i, h_j = 1, ..., R. \tag{9}$$

Note that when the ordinal difference between two data points increases, the extent of distance enlargement (the second term in W_{ij}) increases accordingly.

By using Laplace regulizer, $\Omega(\|f\|_K)$, in concert with other regularizers, we obtain the Laplacian-regularized KCORF model, which loss function is given by (5) but where

$$\Omega_K(\theta, \alpha, \tau) = \lambda_1 \Omega(\|f\|_K) + \lambda_2 \|\alpha\|^2 + \lambda_3 \|\theta\|^2. \tag{10}$$

The additional regularization based on graph Laplacian incorporates geometric structure in the kernel-based regularization of the KCORF model. In other words, it constrains the conditional probability distribution $P(h|f(x), \theta, \alpha, \tau)$ to vary smoothly along the geodesics in the intrinsic geometry of $P(x, h)$, which is ignored in the CORF and standard CRF models.

3.3 Lap-KCORF: Learning and Inference

The final objective function of Lap-KCORF is obtained by plugging the kernel feature functions defined in (7), and the regularization term in (10), into (5). Note that, in addition to Laplacian regularization, we also use the L_2 regularizer for the kernel weights α, in order to avoid diverging solutions. The parameters b and σ are re-parametrized, as explained in [8], in order to arrive at the unconstrained minimization problem. The minimization of such objective is achieved using the quasi-Newton limited-BFGS method (see [8] for gradient derivations). We now describe briefly the learning strategy. Initially, we use KLPP to set the kernel weights α. Then, we set the edge parameters $u = 0$ to form a static ranking model that treats each node independently. After learning the node and kernel parameters $\{b, \sigma, \alpha, \tau\}$, we optimize the model w.r.t. u while holding the other parameters fixed. In the final step, we optimize all model parameters together. Once the parameters of the Lap-KCORF model are found, the inference of test sequences is carried out using Viterbi decoding.

[2] Note that this graph is different from the one defined in CORF in Sec. 2

4 Recognition of AU temporal segments

In this section, we first adapt the proposed Lap-KCORF model to the target task. We then introduce Composite Histogram Intersection (CHI) kernel that we employ in the Lap-KCORF model.

4.1 Lap-KCORF for recognition of AU temporal segments

Recognition of AU temporal segments (neutral, onset, apex and offset) is usually cast as a four-class classification problem. In addition to categorical labels $(\text{neutral} = 1, \text{onset} = 2, \text{apex} = 3, \text{offset} = 4)$, we also assign ordinal labels to the temporal segments $(\text{neutral} = 1 \prec \text{onset}, \text{offset} = 2 \prec \text{apex} = 3)$. To incorporate this into the Lap-KCORF model, we need to lower its assumption that all classes have different and monotonically increasing ranking scores. This is attained by re-defining its node features as

$$P(h = \text{onset}|f_s(x)) = P(h = \text{offset}|f_s(x)) = P(f_s(x) \in [b_0, b_1)). \quad (11)$$

Note that with such-defined node features, the Lap-KCORF model can perform classification of neutral and apex solely based on their ordinal scores. However, to differentiate between onset and offset, Lap-KCORF has to rely completely on its dynamic features, where the transition matrix u and the intensity of the appearance change, measured in the ordinal space by $f_d(x_t, x_{t-1})$, play key role in discriminating between onset and offset. Fig.1 illustrates the importance of modeling the sign in dynamic features for discerning the two phases of equal ordinal score.

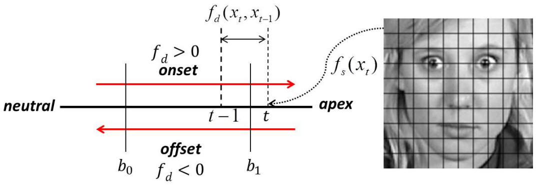

Fig. 1. Modeling of AUs' temporal segments in the ordinal space of Lap-KCORF.

4.2 Composite Histogram Intersection (CHI) Kernel

For classification of the temporal segments using the Lap-KCORF, we propose the CHI kernel derived from the widely used Histogram Intersection (HI) kernel [13]. The HI kernel is specifically designed for measuring similarity between two histograms. Formally, we are given two image histograms x_i and x_j, both containing M bins, where the b-th bin has the value x_i^b and x_j^b, respectively. If we assume that x_i and x_j have the same size, i.e., $\sum_{b=1}^{M}\mathrm{x}_i^b = \sum_{b=1}^{M}\mathrm{x}_j^b$, then the HI kernel is given by: $\mathrm{k}(\mathrm{x}_i, \mathrm{x}_j) = \sum_{b=1}^{M}\min\left\{\mathrm{x}_i^b, \mathrm{x}_j^b\right\}$. To compute the histograms, we employ Local Binary Patterns (LBPs) since they have been shown to be effective in the task of AU detection [14]. Specifically, we first align facial images, and then divide each image into 10x10 equally sized non-overlapping regions. LBP histograms are then extracted from each region, resulting in a 59-D feature vector per each region.

To combine information from multiple local regions, we propose the Composite Histogram Intersection (CHI) kernel:

$$k_{chi}(\mathrm{x}_i,\mathrm{x}_j) = \sum_{r=1}^{R}\beta_r \mathrm{k}_\mathrm{r}(\mathrm{x}_i,\mathrm{x}_j), \quad \beta_r \geq 0, \quad \sum_{r=1}^{R}\beta_r = 1, \tag{12}$$

where β_r is learned from training data to reflect the relevance of region r, to which we apply the HI kernel $k_r(\cdot,\cdot)$. The positiveness constraint ensures that $k_{chi}(\cdot,\cdot)$ is positive definite, and the unitary constraint is necessary to avoid diverging solutions. To avoid constrained optimization of the kernel parameters β, we introduce re-parameterization: $\beta_r = Z_\tau^{-1}e^{\tau_r}$, where Z_τ is the normalization const. The CHI kernel automatically finds facial regions important for the classification of the AUs' temporal segments, and discards the irrelevant ones. This may help to reduce overfitting, and, thus, improve predictive accuracy of the classifier.

5 Experiments

We evaluated the proposed approach using the MMI facial expression database (MMI-db)[15], parts I and II. Specifically, we used videos depicting facial expressions of single AU activation, performed by different subjects. Furthermore, in this paper we report results only for AUs from the upper face (i.e., AU1, AU2, AU3, AU4, AU5, AU6, AU7, AU43, AU45 and AU46). The activation of each AU is manually coded per frame into one of four temporal segments (neutral, onset, apex or offset), and it is provided by the db creators. We refer our reader to [4] for more details about the db, and the AUs that we address in this paper.

We trained the proposed Lap-KCORF model for each AU separately using the corresponding image sequences. The parameter learning was done as explained in Sec.3.3. As input features, we used 5x10 LBP histograms computed from the upper face of the aligned training images (see Sec.4.2). Furthermore, we initialized the weights β_r of the CHI kernel by assuming uniform prior. To reduce the computational cost of the Lap-KCORF model, without significantly reducing the model's performance, we set the maximum number of the kernel bases to #300. The bases were sampled uniformly at random from training examples of each temporal segment.

We compared the performance of Lap-KCORF to that achieved by Lap-CORF[9] in the target task. Since Lap-CORF[9] is a regularized version of the base CORF model proposed in [8], we do not include the results for CORF in

Table 1. F1-score for each AU.

Method	AU1	AU2	AU4	AU5	AU6	AU7	AU43	AU45	AU46	Av.
SVM-HMM[4]	0.65	**0.69**	0.54	0.45	0.58	0.34	**0.72**	**0.78**	0.29	0.56
Lap-CORF[9]	0.67	0.64	0.52	0.55	0.59	0.54	0.57	0.65	0.48	0.58
Lap-KCORF	**0.70**	0.68	**0.58**	**0.60**	**0.63**	**0.59**	0.69	0.71	**0.62**	**0.65**

this paper. For Lap-CORF, the values of the 50 histograms of each image were concatenated to form a vector. The training set of such vectors was pre-processed by applying PCA to reduce its dimensionality (∼25) by preserving 98% of energy. Lap-KCORF used full histogram features. We also show the performance of the Hybrid SVM-HMM [4] model, the state-of-the-art approach to automatic recognition of AUs' temporal segments, which is based on geometric features. Finally, in all our experiments we applied 5-fold cross validation procedure[3], where each fold contained image sequences of different subjects. We report the accuracy using the F-1 measure, defined as $2pr/(p+r)$, where p and r represent obtained precision and recall, respectively.

Table 2. F1-score for each temporal segment.

Method	neutral	onset	apex	offset
SVM-HMM[4]	**0.78**	0.45	0.57	0.44
Lap-CORF[9]	0.65	0.48	0.68	0.50
Lap-KCORF	0.72	**0.53**	**0.79**	**0.54**

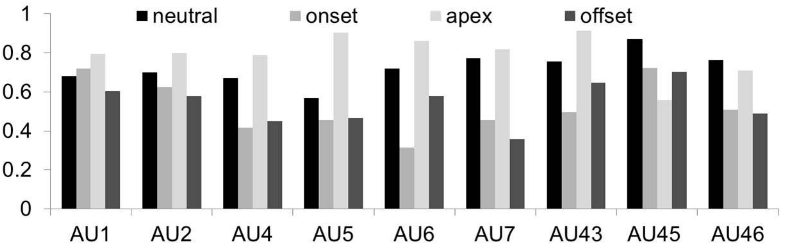

Fig. 2. Lap-KCORF: F1 score for temporal segments of different AUs.

Table 1 shows the average performance of recognition of temporal segments of different AUs. SVM-HMM and Lap-CORF perform similarly on average. It

[3] We used three folds for training, one for validation - to find the regularization parameters, and one for testing.

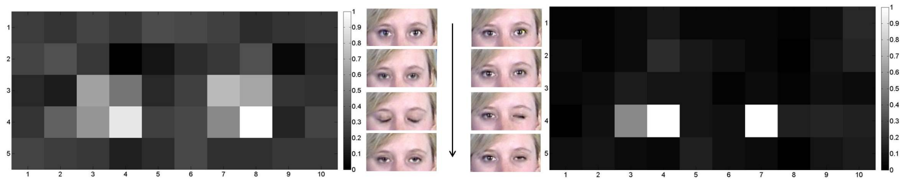

Fig. 3. The weights of the CHI kernel learned for AU45 (*left*) and AU46 (*right*).

is interesting to note that parametric Lap-CORF performs this well having that it uses linear feature functions, in contrast to its kernel counterparts, i.e., SVM-HMM and Lap-KCORF. Although SVM-HMM performs better than the proposed Lap-KCORF on certain AUs, the latter model exhibits better performance on average. This is attributed in part to its superior recognition of AU7 and AU46. The same can be observed from the results per segment shown in Table 2. Note that both Lap-CORF and Lap-KCORF outperform SVM-HMM on all temporal segments except neutral, which signals that these models are better suited for modeling the *dynamics* of an AU activation. Since all these models model the dynamics by means of the parametric transition matrix, we attribute the better performance achieved by Lap-K/CORF to their modeling of the static ordinal constraints (important for the apex-segment recognition). Furthermore, the better performance of Lap-KCORF compared to that of Lap-CORF is in part due to Lap-KCORF's modeling of the temporal dynamics in the ordinal space as explained in Sec. 4.1 (crucial for differentiation between the onset and the offset phases), and in part due to the proposed CHI kernel, which selects 'good' features for the target task. Finally, it is interesting to note from Fig.5 that Lap-KCORF recognizes well the apex of AUs' activation in all cases except for AU45. We inspected the data of this AU, and found that there were only few examples of the apex present. Hence, Lap-KCORF did not have sufficient support of the kernel bases of the apex, which affected its performance in this particular task.

Fig.3 depicts the relevance (measured by the values β_r of the CHI kernel) of facial regions for the recognition of AU45 (blink) and AU46 (wink).The reason why in AU46 we have 'relevant' regions on both sides of the face is that we used examples of AU46L (left wink) and AU46R (right blink) together to train the model. Note that in the case of AU46 we have much sparser β's. This is due to the fact that in AU46 the closure of the eye, which is annotated as apex, lasts much longer than in AU45, and, thus, the model assigns more weight to the region where the eye stayed closed. In conclusion, these maps can be used to further analyze the dynamics of AU activations.

6 Conclusion

We proposed the Lap-KCORF model for the recognition of AUs' temporal segments. We also proposed the Composite Histogram Intersection kernel for automatic learning of relevance of the facial regions for the target task. Our experimental results suggest that ordinal relations between AUs' temporal segments play an important role in the recognition task, in addition to their temporal relations. The proposed Lap-KCORF model can also be applied to AU intensity estimation, the problem addressed in [3]. This is part of our ongoing research.

Acknowledgments. This material is based upon work supported by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB), and by the National Science Foundation under Grant No. IIS 0916812.

References

1. Pantic, M.: Machine analysis of facial behaviour: Naturalistic and dynamic behaviour. Philosophical Transactions of Royal Society B **364** (2009) 3505–3513
2. Ekman, P., Friesen, W., Hager, J.: Facial Action Coding System (FACS): Manual. A Human Face (2002)
3. Mahoor, M., Cadavid, S., Messinger, D., Cohn, J.: A framework for automated measurement of the intensity of non-posed facial action units. CVPRW (2009) 74–80
4. Valstar, M.F., Pantic, M.: Fully automatic recognition of the temporal phases of facial actions. IEEE Transactions on Systems, Man and Cybernetics **42** (2012) 28–43
5. Koelstra, S., Pantic, M., Patras, I.: A dynamic texture based approach to recognition of facial actions and their temporal models. IEEE PAMI **32** (2010) 1940–1954
6. Pantic, M., Patras, I.: Detecting facial actions and their temporal segments in nearly frontal-view face image sequences. Proc. of IEEE Int'l Conf. Systems, Man and Cybernetics (2005) 3358–3363
7. Pantic, M., Patras, I.: Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. IEEE Transactions on Systems, Man and Cybernetics - Part B **36**(2) (2006) 433–449
8. Kim, M., Pavlovic, V.: Structured output ordinal regression for dynamic facial emotion intensity prediction. ECCV (2010) 649–662
9. Rudovic, O., Pavlovic, V., Pantic, M.: Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. CVPR (2012) In press.
10. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. (2001) 282–289 ICML.
11. He, X., Niyogi, P.: Locality Preserving Projections. NIPS (2004)
12. Lafferty, J., Zhu, X., Liu, Y.: Kernel conditional random fields: representation and clique selection. ICML '04 (2004) 64–
13. Barla, A., Odone, F., Verri, A.: Histogram intersection kernel for image classification. ICIP 2003 **3** (2003) III – 513–16 vol.2
14. Jiang, B., Valstar, M.F., Pantic, M.: Action unit detection using sparse appearance descriptors in space-time video volumes. FG'11 (2011) 314–321
15. Pantic, M., Valstar, M.F., Rademaker, R., Maat, L.: Web-based database for facial expression analysis. In: ICME'05. (July 2005) 317–321