Pairwise Conditional Random Forests for Facial Expression Recognition

Arnaud Dapogny[1]
arnaud.dapogny@isir.upmc.fr

Kevin Bailly[1]
kevin.bailly@isir.upmc.fr

Séverine Dubuisson[1]
severine.dubuisson@isir.upmc.fr

[1] Sorbonne Universités, UPMC Univ Paris 06, CNRS, ISIR UMR 7222, 4 place Jussieu 75005 Paris

Abstract

Facial expression can be seen as the dynamic variation of one's appearance over time. Successful recognition thus involves finding representations of high-dimensional spatio-temporal patterns that can be generalized to unseen facial morphologies and variations of the expression dynamics. In this paper, we propose to learn Random Forests from heterogeneous derivative features (e.g. facial fiducial point movements or texture variations) upon pairs of images. Those forests are conditioned on the expression label of the first frame to reduce the variability of the ongoing expression transitions. When testing on a specific frame of a video, pairs are created between this current frame and the previous ones. Predictions for each previous frame are used to draw trees from Pairwise Conditional Random Forests (PCRF) whose pairwise outputs are averaged over time to produce robust estimates. As such, PCRF appears as a natural extension of Random Forests to learn spatio-temporal patterns, that leads to significant improvements over standard Random Forests as well as state-of-the-art approaches on several facial expression benchmarks.

Introduction

In the last decades, automatic facial expression recognition (FER) has attracted an increasing attention, as it is a fundamental step of many applications such as human-computer interaction, or assistive healthcare technologies. A good survey covering FER can be found in [23]. There exists many impediments to successful deciphering of facial expressions, among which the large variability in morphological and contextual factors as well as the subtlety of low-intensity expressions. Because using dynamics of the expression helps disentangle those factors [4], most recent approaches aim at exploiting the temporal variations in videos rather than trying to perform recognition on still images.

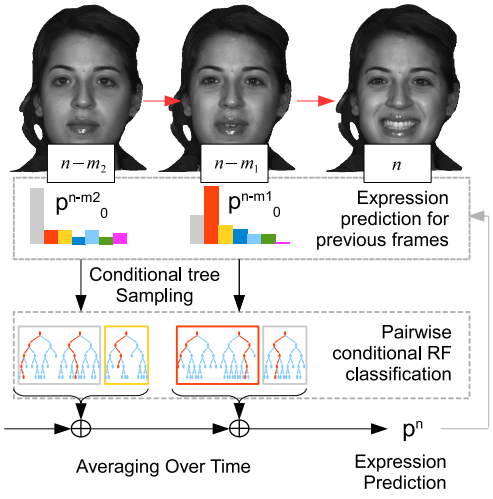

Figure 1. Flowchart of our PCRF FER method. When evaluating a video frame indexed by n, dedicated pairwise trees are drawn conditionally to expression probability distributions from previous frames $n-m_1, n-m_2, ...$ The subsequent predictions are averaged over time to output an expression probability distribution for the current frame. This prediction can be used as a tree sampling distribution for subsequent frame classification.

Nevertheless, as a matter of fact, the semantics of acted facial events is composed of successive *onset*, *apex* and *offset* phases that may occur with various offsets and at different paces. Spontaneous behaviors can however be much more difficult to analyse in terms of such explicit sequence. Hence, there is no consensus on either how to effectively extract suitable representations from those high dimensional video patterns, or on how to combine those representations in a flexible way to generalize to unseen data and possibly unseen temporal variations. Common approaches employ spatio-temporal descriptors defined on fixed-size win-

dows, optionally at multiple resolutions. Examples of such features include the so-called LBP-TOP [25] and HoG3D [10] descriptors, which are spatio-temporal extensions of LBP and HoG features respectively. Authors in [14] use histograms of local phase and orientations. However, such representations may lack the capacity to generalize to facial events that differ from training data on the temporal axis.

Other approaches aim at establishing relationships between high-level features and a sequence of latent states. Wang *et al.* [19] integrate temporal interval algebra into a Bayesian Network to capture complex relationships among facial muscles. Walecki *et al.* [17] introduce a variable-state latent Conditional Random Field that can automatically select the optimal latent states for individual expression videos. Such approaches generally require explicit dimensionality reduction techniques such as PCA or K-means clustering for training. In addition, training at the sequence level considerably reduces the quantity of available training and testing data as compared to frame-based approaches.

In early work [6] we obtained promising results for acted video FER by learning a restricted transition model that we fused with a static one. However, this model lacks the capacity to generalize to more complex spontaneous FER scenarios where one expression may quickly follow another one. Furthermore, because the transition and static classifiers are applied as two separate tasks, it can not integrate low-level correlations between static and spatio-temporal patterns. In this paper, we introduce the Pairwise Conditional Random Forest (PCRF) algorithm, which is a new formulation for training trees using low-level heterogeneous static (spatial) and dynamic (spatio-temporal derivative) features within the RF framework. Conditional Random Forests were used by Dantone *et al.* [5] as well as Sun *et al.* [15] in the field of facial alignment and human pose estimation, respectively. They generated collections of trees for specific, quantized values of a global variable (such as head pose [5] and body torso orientation [15]) and used prediction on this global variable to draw dedicated trees, resulting in more accurate predictions. In our case, we propose to condition pairwise trees upon specific expression labels to reduce the variability of ongoing expression transitions from the first frame of the pair to the other one (Figure 1). When evaluating a video, each previous frame of the sequence is associated with the current frame to give rise to a pair. Pairwise trees are thus drawn from the dedicated PCRFs w.r.t. a prediction for the previous frame. Finally, predictions outputted for each pair are averaged over time to produce a robust prediction for the current frame. The contributions of this work are thus three-fold:

- A method for training pairwise random trees upon high-dimensional heterogeneous static and spatio-temporal derivative feature templates, with a conditional formulation that reduces the variability.
- An algorithm that is a natural extension of the static RF averaging, that consists in averaging over time pairwise predictions to flexibly handle temporal variations.
- A framework that performs FER from video that can work on low-power engines thanks to an efficient implementation using integral feature channels.

The rest of the paper is organized as follows: in Section 1 we describe our adaptation of the RF framework to learn expression patterns on still images from high-dimensional, heterogeneous features. In Section 2 we present the PCRF framework to learn temporal patterns of facial expressions. In Section 3 we show how our PCRF algorithm improves the accuracy on several FER datasets as compared to a static approach. We also show how our method outperforms the state-of-the-art approaches and report its ability to run in real-time. Finally, we give concluding remarks on our PCRF for FER and discuss perspectives.

1. Random Forests for FER

1.1. Facial expression prediction

Random Forests (RFs) [2] is a popular learning framework introduced in the seminal work of Breiman [2]. They have been used to a significant extent in computer vision and for FER tasks in particular due to their ability to nicely handle high-dimensional data such as images or videos as well as being naturally suited for multiclass classification tasks. They combine random subspace methods for feature sampling and bagging in order to provide performances similar to the most popular machine learning methods such as SVM or Deep Neural Networks.

RFs are classically built from the combination of T decision trees grown from bootstraps sampled from the training dataset. In our implementation, we downsample the majority classes within the bootstraps in order to enforce class balance. As compared to other methods for balancing RF classifiers (*i.e.* class weighting and upsampling of the minority classes), downsampling leads to similar results while substantially reducing the computational cost, as training is performed on smaller data subsets.

Individual trees are grown using a greedy procedure that involves, for each node, the measure of an impurity criterion $H_{\phi,\theta}$ (which is traditionally either defined as Shannon entropy or Gini impurity measurement) relatively to a partition of the images x with label $l \in \mathcal{L}$, that is induced by candidate binary split functions $\{\phi, \theta\} \in \Phi$. More specifically, we use multiple parametric feature templates to generate multiple heterogeneous split functions, that are associated with a number of thresholds θ. In what follows, by abuse of notations we will refer to $\phi^{(i)}$ as the i^{th} feature template and $k^{(i)}$ as the number of candidates generated from this template. The "Best" binary feature among all features from

the different templates (*i.e.* the one that minimizes the impurity criterion $H_{\phi,\theta}$) is set to produce a data split for the current node. Then, those steps are recursively applied for the left and right subtrees with accordingly rooted data until the label distribution at each node becomes homogeneous, where a leaf node can be set. This procedure for growing trees is summarized in Algorithm 1.

Algorithm 1 Tree Growing algorithm `treeGrowing`

input: images x with labels l, root node n, number of candidate features $\{k^{(i)}\}_{i=1,2,3}$ for templates $\{\phi^{(i)}\}_{i=1,2,3}$

```
if image labels are homogeneous with value l_0 then
    set node as terminal, with probabilities p_t to 1 for l_0, 0 elsewhere
else
    generate an empty set of split candidates Φ
    for all feature templates i do,
        generate a set Φ^(i) of k^(i) candidates {φ^(i), θ}
        Φ ← Φ ∪ Φ^(i)
    end for
    for {φ, θ} ∈ Φ do
        compute the impurity criterion H_{φ,θ}(x)
    end for
    split data w.r.t. arg min_{φ,θ}{H_{φ,θ}(x)} in left and right subsets x_l and x_r
    create left n_l and right n_r children of node n
    call treeGrowing(x_l,n_l,{k^(i)}_{i=1,2,3})
    call treeGrowing(x_r,n_r,{k^(i)}_{i=1,2,3})
end if
```

During evaluation, an image x is successively rooted left or right of a specific tree t according to the outputs of the binary tests, until it reaches a leaf node. The tree thus returns a probability $p_t(l|x)$ which is set to either 1 for the represented class, or to 0. Prediction probabilities are then averaged among the T trees of the forest (Equation (1)).

$$p(l|x) = \frac{1}{T}\sum_{t=1}^{T} p_t(l|x) \tag{1}$$

Note that the robustness of the RF prediction framework comes from (a) the strength of individual trees and (b) the decorrelation between those trees. By growing trees from different bootstraps of available data and with the random subspace algorithm (e.g. examining only a subset of features for splitting each node) we generate individually weaker, but less correlated trees that provide better combination predictions than standard CART or C4.5 procedures.

1.2. Heterogeneous feature templates

Feature templates $\phi^{(i)}$ include both geometric (*i.e.* computed from previously aligned facial feature points) and appearance features. Each of these templates have different input parameters that are randomly generated during training by uniform sampling over their respective variation range. Also, during training, features are generated along with a set of candidate thresholds θ to produce binary split candidates. In particular, for each feature template $\phi^{(i)}$, the upper and lower bounds are estimated from training data beforehand and candidate thresholds are drawn from a uniform distribution in the range of these values.

We use two different geometric feature templates which are generated from the set of facial feature points $f(x)$ aligned on image x with the SDM tracker [20]. The first geometric feature template $\phi^{(1)}_{a,b}$ is the euclidian distance between feature points f_a and f_b, normalized w.r.t. inter-ocular distance $iod(f(x))$ for scale invariance (Equation 2).

$$\phi^{(1)}_{a,b}(x) = \frac{||f_a - f_b||_2}{iod(f)} \tag{2}$$

Because the feature point orientation is discarded in feature $\phi^{(1)}$ we use the angles between feature points f_a, f_b and f_c as our second geometric feature $\phi^{(2)}_{a,b,c,\lambda}$. In order to ensure continuity for angles around 0, we use the cosine and sine instead of the raw angle value. Thus, $\phi^{(2)}$ outputs either the cosine or sine of angle $\widehat{f_a f_b f_c}$, depending on the value of the boolean parameter λ (Equation (3)):

$$\phi^{(2)}_{a,b,c,\lambda}(x) = \lambda\cos(\widehat{f_a f_b f_c}) + (1-\lambda)\sin(\widehat{f_a f_b f_c}) \tag{3}$$

We use Histogram of Oriented Gradients (HoG) as our appearance features for their descriptive power and robustness to global illumination changes. In order to ensure fast feature extraction, we use integral feature channels as introduced in [8]. First, we rotate the image x in order to align the inter-ocular axis on the horizontal axis. We then compute horizontal and vertical gradients on the rotated image and use these to generate 9 feature channels, the first one containing the gradient magnitude. The 8 remaining channels correspond to a 8-bin quantization of the gradient orientation. Finally, integral images are computed from these feature maps to output 9 channels. Thus, we define the appearance feature template $\phi^{(3)}_{\tau,ch,s,\alpha,\beta,\gamma}$ as an histogram computed over channel ch within a window of size s normalized w.r.t. the inter-ocular distance. Such histogram is evaluated at a point defined by its barycentric coordinates α, β and γ w.r.t. vertices of a triangle τ defined over feature points $f(x)$. Also, storing the gradient magnitude within the first channel allows to normalize the histograms as in standard HoG implementation. Thus, HoG features can be computed very efficiently by using only 4 access to the integral channels (plus normalization).

2. Learning temporal patterns with PCRF

2.1. Learning PCRF with heterogeneous derivative feature templates

In this section we explain how we adapt the aforementioned RF framework to take into account the dynamics of the expression. We now consider pairs of images (x',x) to train trees t that aim at outputting probabilities $p_t(l|x', x, l')$ of observing label $l(x) = l$ given image x' and subject to $l(x') = l'$, as illustrated in Figure 2.

More specifically, for each tree t among the T trees of a RF dedicated to transitions starting from expression label l', we randomly draw a fraction of subjects $\tilde{\mathcal{S}} \subset \mathcal{S}$. Then, for each subject $s \in \tilde{\mathcal{S}}$ we randomly draw images x'_s that specifically have label l'. We also draw images x_s of every label l and create as many pairs (x'_s, x_s) with label l. Note that the two images of a pair do not need to belong to the same video. Instead, we create pairs from images sampled across different sequences for each subject to cover all sorts of ongoing transitions. We then balance the pairwise bootstrap by downsampling the majority class w.r.t. the pairwise labels. Eventually, we construct tree t by calling algorithm 1. Those steps are summarized in Algorithm 2.

Algorithm 2 Training a PCRF

input: images x with labels l, number of candidate features $\{k^{(i)}\}_{i=1,...,6}$ for templates $\{\phi^{(i)}\}_{i=1,...,6}$

```
for all l' ∈ L do
    for t = 1 to T do
        randomly draw a fraction S̃ ⊂ S of subjects
        pairs ← {}
        for all s ∈ S̃ do
            draw samples x'_s with label l'
            draw samples x_s for each label l
            create pairwise data (x'_s, x_s) with label l
            add element (x'_s, x_s) to pairs
        end for
        balance bootstrap pairs with downsampling
        create new root node n
        call treeGrowing(pairs,n,{k^(i)}_{i=1,...,6})
    end for
end for
```

Candidates for splitting the nodes are generated from an extended set of 6 feature templates $\{\phi^{(i)}\}_{i=1,...,6}$, three of which being the static features described in Section 1, that are applied to the second image x of the pair (x', x), for which we want to predict facial expressions. The three remaining feature templates are dynamic features defined as the derivatives of static templates $\phi^{(1)}$, $\phi^{(2)}$, $\phi^{(3)}$ with the exact same parameters. Namely, we have:

$$\begin{cases} \phi^{(1)}_{a,b}(x', x) & = \phi^{(1)}_{a,b}(x) \\ \phi^{(2)}_{a,b,c,\lambda}(x', x) & = \phi^{(2)}_{a,b,c,\lambda}(x) \\ \phi^{(3)}_{\tau,ch,s,\alpha,\beta,\gamma}(x', x) & = \phi^{(3)}_{\tau,ch,s,\alpha,\beta,\gamma}(x) \\ \phi^{(4)}_{a,b}(x', x) & = \phi^{(1)}_{a,b}(x) - \phi^{(1)}_{a,b}(x') \\ \phi^{(5)}_{a,b,c,\lambda}(x', x) & = \phi^{(2)}_{a,b,c,\lambda}(x) - \phi^{(2)}_{a,b,c,\lambda}(x') \\ \phi^{(6)}_{\tau,ch,s,\alpha,\beta,\gamma}(x', x) & = \phi^{(3)}_{\tau,ch,s,\alpha,\beta,\gamma}(x) - \phi^{(3)}_{\tau,ch,s,\alpha,\beta,\gamma}(x') \end{cases} \tag{4}$$

As in Section 1, thresholds for the derivative features $\phi^{(4)}$, $\phi^{(5)}$, $\phi^{(6)}$ are randomly drawn from uniform distributions with new dynamic template-specific ranges estimated from the pairwise dataset beforehand. Also note that, as compared to a static RF, a PCRF model is extended with new derivative features that are estimated from a pair of images. When applied on a video, predictions for several pairs are averaged over time in order to produce robust estimates of the probability predictions.

2.2. Model averaging over time

We denote by $p^n(l)$ the prediction probability of label l for a video frame x^n . For a purely static RF classifier this probability is given by Equation (5):

$$p^n(l) = \frac{1}{T} \sum_{t=1}^{T} p_t(l|x^n) \tag{5}$$

In order to use spatio-temporal information, we apply pairwise RF models to pairs of images (x^m, x^n) with $\{x^m\}_{m=n-1,...,n-N}$ the previous frames in the video. Those pairwise predictions are averaged over time to provide a new probability estimate p^n that takes into account past observations up to frame n. Thus, if we do not have prior information for those frames the probability p^n is:

$$p^n(l) = \frac{1}{NT} \sum_{m=n-1}^{n-N} \sum_{t=1}^{T} p_t(l|x^m, x^n) \tag{6}$$

In what follows, Equation (6) will be refered to as the *full* model. Trees from the full model are likely to be stronger that those of the static one since they are grown upon an extended set of features. Likewise, the correlation between the individual trees is also lower thanks to the new features and the fact that they will be evaluated on multiple, distinct pairs when averaging over time. However, spatio-temporal information can theoretically not add much to the accuracy if the variability of the pairwise data points is too large.

In order to decrease this variability, we assume that there exists a probability distribution $p^m_0(l')$ to observe the discrete expression label l' at frame m. Note that probabilities p_0 can be set to purely static estimates (which is necessarily the case for the first video frames) or dynamic predictions estimated from previous frames. A comparison between

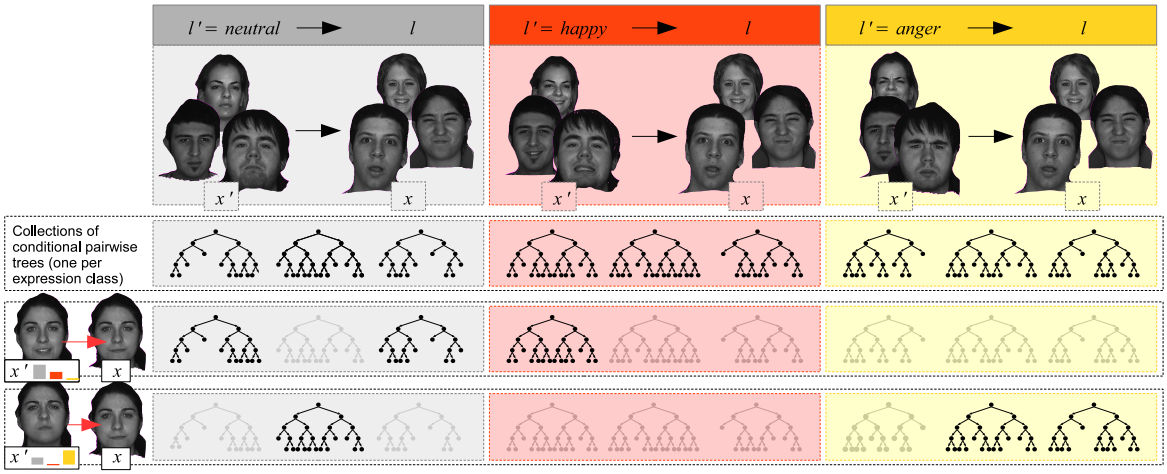

Figure 2. Expression recognition from pairs of images using PCRF. Expression probability predictions of previous images are used to sample trees from dedicated pairwise tree collections (one per expression class) that are trained using subsets of the (pairwise) training dataset, with only examples of ongoing transitions from a specific expression towards all classes. The resulting forest thus outputs an expression probability for a specific pair of images.

those approaches can be found in Section 3.3. In such a case, for frame m, pairwise trees are drawn conditionally to distributions p_0^m, as shown in Figure 2. More specifically, for each expression label l' we randomly select $\mathcal{N}(l')$ trees over a PCRF model dedicated to transitions that start from expression label l', trained with the procedure described in Section 2.1. Equation (6) thus becomes:

$$p^n(l) = \frac{1}{NT} \sum_{m=n-1}^{n-N} \sum_{l' \in \mathcal{L}} \sum_{t=1}^{\mathcal{N}(l')} p_t(l|x^m, x^n, l') \qquad (7)$$

Where $\mathcal{N}(l') \approx T.p_0^m(l')$ and $T = \sum_{l' \in \mathcal{L}} \mathcal{N}(l')$ being the number of trees dedicated to the classification of each transition, which can be set in accordance with CPU availability. In our experiments, we will refer to Equation (7) as the *conditional* model. This formulation reduce the variability of the derivative features for each specialized pairwise RF. Section 3 shows that using PCRF models leads to significant improvements over both static and full models.

3. Experiments

In this section, we report comparisons between different classification models on two well-known FER databases, the Extended Cohn-Kanade and BU-4DFE databases. Furthermore, in order to evaluate the capabilities of the learned models to generalize on new, potentially more complicated FER scenarios, we report classification results for cross-database evaluation on two spontaneous databases, namely the FG-NET FEED and BP4D databases. We highlight that our conditional formulation of dynamic integration substantially increases the recognition accuracy on such tasks. Finally, we show the real-time capacitbility of our system.

3.1. Databases

The CK+ or **Extended Cohn-Kanade database [12]** contains 123 subjects, each one associated with various numbers of expression records. Those records display a very gradual evolution from a *neutral* class towards one of the 6 universal facial expressions (*anger*, *happiness*, *sadness*, *fear*, *digust* and *surprise*) plus the nonbasic expression *contempt*. Expressions are acted with no head pose variation and their duration is about 20 frames. From this dataset we use 309 sequences, each one corresponding to one of the six basic expressions, and use the three first and last frames from these sequences for training. We did not include sequences labelled as *contempt* because CK+ contains too few subjects performing *contempt* and other expressions to train the pairwise classifiers.

The BU-4DFE database [22] contains 101 subjects, each one displaying 6 acted facial expressions with moderate head pose variations. Expressions are still prototypical but they are generally exhibited with much lower intensity and greater variability than in CK+, hence the lower baseline accuracy. Sequences duration is about 100 frames. As the database does not contain frame-wise expression annotations, we manually select neutral and apex of expression frames for training.

The BP4D database [24] contains 41 subjects. Each subject was asked to perform 8 tasks, each one supposed to give rise to 8 spontaneous facial expressions (*anger*, *happiness*, *sadness*, *fear*, *digust*, *surprise*, *embarrassment* or *pain*). In [24] the authors extracted subsequences of about 20 seconds for manual FACS annotations, arguing that these subsets contains the most expressive behaviors.

The FG-NET FEED database [18] contains 19 subjects, each one recorded three times while performing 7 spontaneous expressions (the six universal expressions, plus *neutral*). The data contain low-intensity emotions, short expression displays, as well as moderate head pose variations.

3.2. Experimental Setup

7-class RF (*static*) and PCRF (*full* and *conditional*) classifiers are trained on the CK+ and BU-4DFE datasets using the set of hyperparameters described in Table 1. These parameters were obtained by cross-validation. Note however that extensive testing showed that the values of these hyperparameters had a very subtle influence on the performances. This is due to the complexity of the RF framework, in which individually weak trees (e.g. that are grown by only examining a few features per node) are generally less correlated, still outputting decent predictions when combined altogether. Nevertheless, we report those settings for reproducibility concerns. Also, for a fair comparison between static and pairwise models, we use the same total number of feature evaluations for generating the split nodes. Moreover, for all the models the maximum accuracy was reached for $T \approx 100$ trees. However, we generated large numbers of trees so that the variance in prediction accuracy for the following benchmarks becomes very low over all the runs.

Table 1. Hyperparameters settings

Hyperparameters	value(static)	value(dynamic)
Nb. of $\phi^{(1)}$ features	40	20
Nb. of $\phi^{(2)}$ features	40	20
Nb. of $\phi^{(3)}$ features	160	80
Nb. of $\phi^{(4)}$ features	-	20
Nb. of $\phi^{(5)}$ features	-	20
Nb. of $\phi^{(6)}$ features	-	80
Data ratio per tree	$2/3$	$2/3$
Nb. of thresholds	25	25
total nb. of features	6000	6000
Nb. of trees	500	500

During the evaluation, the prediction is initialized at the first frame using the static classifier. Then, for the full and conditional models, probabilities are estimated for each frame using transitions from previous frames only, bringing us closer to a real-time scenario. However, although it uses transitional features, our system is essentially a frame-based classifier that outputs an expression probability for each separate video frame. This is different from, for example, a HMM, that aims at predicting a probability related to all the video frames. Thus, in order to evaluate our classifier on video FER tasks, we acknowledge correct classification if the maximum probability outputted for all frames corresponds to the ground truth label. This evaluates the capability of our system to retrieve the most important expression mode in a video, as well as the match between the retrieved mode and the ground truth label.

For the tests on CK+ and BU-4DFE databases, both static and transition classifiers are evaluated using the Out-Of-Bag (OOB) error estimate [2]. More specifically, bootstraps for individual trees of both static and pairwise classifiers are generated at the subject level. Thus, during evaluation, each tree is applied only on subjects that were not used for its training. The OOB error estimate is an unbiased estimate of the true generalization error [2] which is faster to compute than Leave-One-Subject-Out or k-fold cross-evaluation estimates. Also, it has been shown to be generally more pessimistic than other error estimates [3], further empathizing the quality of the proposed results.

3.3. Evaluation of PCRF

In order to validate our approach, we compared our conditional model to a purely static model and a full model, for a variety of dynamic integration parameters (the length of the temporal window N and the step between those frames *Step*) on the BU-4DFE database. We also evaluated the interest of using a *dynamic* probability prediction for previous frames (*i.e.* the output of the pairwise classifier for those frames) versus a *static* one. Average results are provided in Figure 3. For CK+ database, sequences are generally too short to show significant differences when varying the temporal window size or the step size. Thus we only report accuracy for full and conditional models with a window size of 30 and a step of 1. Per-expression accuracies and F1-Scores for both Cohn-Kanade and BU-4DFE databases are shown in Figure 4.

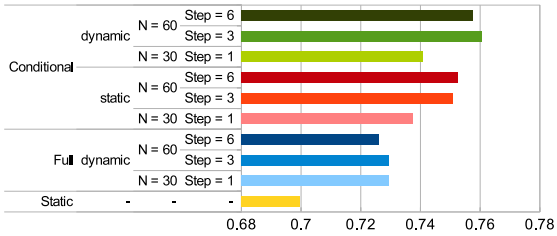

Figure 3. Average accuracy rates obtained for various temporal integration parameters on the BU-4DFE database

We observe that the conditional model significantly outperforms the static model on both CK+ and BU-4DFE databases. This is due to the extra dynamic features that provide both robustness and decorrelation of the individual decision trees, further enhancing their prediction accuracy. Figure 4 shows that the conditional model also outperforms the full model on both databases, which is probably due to the fact that using only a restricted set of ongoing expression transitions for training allows to better capture the variabil-

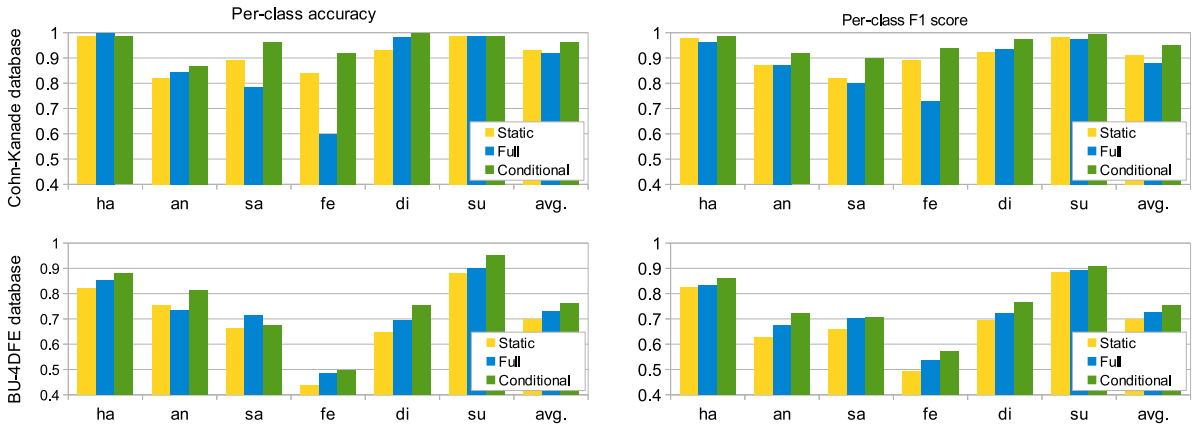

Figure 4. Per-class recognition accuracy rates and F1-scores on CK+ and BU-4DFE databases

ity of the spatio-temporal features for the dedicated pairwise forests. This is particularly true on the CK+ database, where the number of pairwise data points is not enough for the *full* model to capture the variability of all possible ongoing transitions, hence justifying the lower accuracy. This seems particularly relevant for expressions such as *fear* or sadness which are the less represented ones in the dataset. Table 3 also shows that it is better to look backward for more frames in the sequence (N = 60) with less correlation between the frames (*Step* = 3 or 6). Again, such setting allows to take more decorrelated paths in the individual trees, giving a better recombination after averaging over time.

A compilation of comparisons to other state-of-the-art approaches for FER from video sequences can be found in Table 2. On the CK+ dataset, we compare our algorithms with recent work results for FER from video on the same subset of sequences (*i.e.* not including *contempt*). However the comparisons are to be put into perspective as the evaluation protocols differ between the methods. PCRF provides results similar to those reported in [17] as well as in [14, 9, 6], although the latter approaches explicitly consider the last (apex) frame. We also consistently outperform HMM and interval-based DBN presented in [19], although the evaluations protocols are not the same. Furthermore, to the best of our knowledge, our approach gives the best results on the BU-4DFE database for automatic FER from videos using $2D$ information only. Our approach provides better results than the dynamic $2D$ approach [16], as well as the frame-based approach presented in [21]. Recently, Meguid *et al.* [1] obtained impressive results using an original RF/SVM system, from training upon the BU-3DFE database which is a purely static database. They employ a post-classification temporal integration scheme, which we believe may be weaker than using dynamic information at the feature level. Finally, the restricted transition model introduced in our early work [6] face difficulties on spontaneous FER tasks where one expression can quickly succeed to another. Conversely, we show that the method proposed in this paper translates well to such spontaneous scenarios.

Table 2. Comparisons with state-of-the-art approaches. § 7-class expression recognition with contempt label. † results respectively reported for two methods (average-vote) from the paper.

Ck+ database	Protocol	Accuracy
Wang *et al.* [19]	15-fold	$86.3^{\S}$
Shojaeilangari *et al.* [14]	LOSO	94.5
Walecki *et al.* [17]	10-fold	94.5
Khan *et al.* [9]	10-fold	95.3
Dapogny *et al.* [6]	OOB	$96.1 - 94.8^{\dagger}$
This work, RF	OOB	93.2
This work, PCRF	OOB	**96.4**
BU-4DFE database	**Protocol**	**Accuracy**
Xu *et al.* [21]	10-fold	63.8
Sun *et al.* [16]	10-fold	67.0
Dapogny *et al.* [6]	OOB	$72.2 - 75.8^{\dagger}$
Meguid *et al.* [1]	Cross-db	73.1
This work, RF	OOB	70.0
This work, PCRF	OOB	**76.1**
FEED database	**Protocol**	**Accuracy**
Dapogny *et al.* [6]	Cross-db	$53.0 - 53.5^{\dagger}$
Meguid *et al.* [1]	Cross-db	53.7
This work, RF	Cross-db	51.9
This work, PCRF	Cross-db	**57.1**
BP4D database	**Protocol**	**Accuracy**
Dapogny *et al.* [6]	Cross-db	$72.2 - 66.2^{\dagger}$
Zhang *et al.* [24]	Cross-db	71.0
This work, RF	Cross-db	68.6
This work, PCRF	Cross-db	**76.8**

For that matter, Table 2 also reports results for cross-database evaluation (with training on the BU-4DFE database) on the FEED database. In order to provide a

fair comparison between our approach and the one presented in [1], we used the same labelling protocol as them. One can see that the performances of their system are better than those of our static RF model, which can be attributed to the fact that they use a more complex classification and posterior temporal integration flowchart. Nevertheless, our PCRF model provides a substantially higher accuracy, which, again, is likely to be due to the use of spatio-temporal features as well as an efficient conditional integration scheme.

We also performed cross-database evaluation on the BP4D database. Again, for a fair comparison, we used the same protocol as in [24], with training on the BU-4DFE database and using only a subset of the tasks (*i.e.* tasks 1 and 8 corresponding to expression labels *happy* and *disgust* respectively). However, we do not retrain a classifier with a subset of 3 expressions as it is done in [24], but instead use our 7-class static and PCRF models with a forced choice between happiness (probability of class *happiness*) and disgust (probability sum of classes *anger* and *disgust*). Such setting could theoretically increase the confusion in our conditional model, resulting in a lower accuracy. However, as can be seen in Table 2, using dynamic information within the PCRF framework allows to substantially increase the recognition rate as compared to a static RF framework. We also overcome the results reported in [24] by a significant margin, further showing the capability of our approach to deal with complex spontaneous FER tasks. Also note that in [24], the authors used the so-called *Nebulae* $3D$ polynomial volume features which are by far more computationally expensive than our geometric and integral HoG $2D$ features. All in all, we believe our results show that the PCRF approach provides significant improvements over a traditional static classification pipeline that translates very well to more complicated spontaneous FER scenarios, where a single video may contain samples of several expressions.

3.4. Complexity of PCRF

An advantage of using conditional models is that with equivalent parallelization they are faster to train than an hypothetical full model learnt on the whole dataset. According to [11] the average complexity of training a RF classifier with M trees is $\mathcal{O}(MKN\log^2 N)$, with K being the number of features to examine for each node and N the size of (2/3 of) the dataset. Thus if the dataset is equally divided into P bins of size $\tilde{N}$ upon which conditional forests are trained (and such that $N = P\tilde{N}$), the average complexity of learning a conditional model now becomes $\mathcal{O}(MKN\log^2 \tilde{N})$.

Same considerations can be made concerning the evaluation, as trees from the full model are bound to be deeper than those from the conditional models. Table 3 shows an example of profiling a PCRF on one video frame with an averaging over 60 frames and a step of 3 frames. Thus the total number of evaluated trees is $20M$ with M being the number of trees dedicated to classifying each frame.

Table 3. Profiling of total processing time for one frame (in ms)

Step	time (ms)
Facial alignment	10.0
Integral HoG channels computation	2.0
PCRF evaluation (M = 50)	4.8
PCRF evaluation (M = 100)	7.8
PCRF evaluation (M = 300)	19.0

This benchmark was conducted on a Intel Core I7-4770 CPU with 32 Go RAM within a C++/OpenCV environment, without any code parallelization. As such, the algorithm already runs in real-time. Furthermore, evaluations of pairwise classification or tree subsets can easily be parallelized to fit real-time processing requirements on low-power engines such as mobile phones. In addition, the facial alignment step can be performed at more than 300 fps on a smartphone with similar performances using the algorithms from [13].

Conclusion and perspectives

In this paper, we introduced the PCRF framework, which integrates high-dimensional, low-level spatio-temporal information through averaging over time pairwise conditional trees. These trees are drawn by considering previous predictions. We showed that our model can be trained and evaluated efficiently, and leads to a significant increase of performances compared to a static RF. In addition, our method works on real-time without specific optimization schemes, and could be run on low-power architectures such as mobile phones by using an appropriate paralellization. Future works will consist in employing conditional models for pose and occlusion handling to adapt the proposed PCRF framework for "in the wild" FER datasets such as the one in [7]. Furthermore, we would like to investigate applications of PCRF for other video classification/regression problems such as Facial Action Unit intensity prediction or body and hand gesture recognition.

Acknowledgements This work was partially supported by the French National Agency (ANR) in the frame of its Technological Research CONTINT program (JEMImE, project number ANR-13-CORD-0004), and by the Labex SMART (ANR-11-LABX-65) within the Investissements d′Avenir program, under reference ANR-11-IDEX-0004-02.

References

[1] M. Abd El Meguid and M. Levine. Fully automated recognition of spontaneous facial expressions in videos using random forest classifiers. *IEEE Transactions on Affective Computing*, 5:151–154, 2014.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] T. Bylander. Estimating generalization error on two-class datasets using out-of-bag estimates. *Machine Learning*, 48(1-3):287–297, 2002.

[4] J. F. Cohn. Foundations of human computing: facial expression and emotion. In *International Conference on Multimodal Interfaces*, pages 233–238, 2006.

[5] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2578–2585, 2012.

[6] A. Dapogny, K. Bailly, and S. Dubuisson. Dynamic facial expression recognition by static and multi-time gap transition joint classification. *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2015.

[7] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Acted facial expressions in the wild database. *Australian National University, Canberra, Australia, Technical Report TR-CS-11-02*, 2011.

[8] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *British Machine Vision Conference*, 2009.

[9] R. A. Khan, A. Meyer, H. Konik, and S. Bouakaz. Human vision inspired framework for facial expressions recognition. In *IEEE International Conference on Image Processing*, pages 2593–2596, 2012.

[10] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference*, pages 995–1004, 2008.

[11] G. Louppe. *Understanding Random Forests: From Theory to Practice*. PhD thesis, University of Liège.

[12] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, pages 94–101, 2010.

[13] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1685–1692, 2014.

[14] S. Shojaeilangari, W.-Y. Yau, J. Li, and E.-K. Teoh. Multi-scale analysis of local phase and local orientation for dynamic facial expression recognition. *Journal ISSN*, 1:1–10, 2014.

[15] M. Sun, P. Kohli, and J. Shotton. Conditional regression forests for human pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3394–3401, 2012.

[16] Y. Sun and L. Yin. Facial expression recognition based on 3D dynamic range model sequences. In *European Conference on Computer Vision*, pages 58–71. 2008.

[17] R. Walecki, O. Rudovic, V. Pavlovic, and M. Pantic. Variable-state latent conditional random fields for facial expression recognition and action unit detection. *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–8, 2015.

[18] F. Wallhoff. Database with facial expressions and emotions from technical university of munich (feedtum). http://cotesys.mmk.e-technik.tu-muenchen.de/waf/fgnet/feedtum.html, 2006.

[19] Z. Wang, S. Wang, and Q. Ji. Capturing complex spatio-temporal relations among facial muscles for facial expression recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3422–3429, 2013.

[20] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 532–539, 2013.

[21] L. Xu and P. Mordohai. Automatic facial expression recognition using bags of motion words. In *British Machine Vision Conference*, pages 1–13, 2010.

[22] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2008.

[23] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.

[24] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard. BP4D-spontaneous: a high-resolution spontaneous 3D dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014.

[25] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.