Learning Bases of Activity for Facial Expression Recognition

Evangelos Sariyanidi, Hatice Gunes, and Andrea Cavallaro

***Abstract*—The extraction of descriptive features from sequences of faces is a fundamental problem in facial expression analysis. Facial expressions are represented by psychologists as a combination of elementary movements known as action units: each movement is localised and its intensity is specified with a score that is small when the movement is subtle and large when the movement is pronounced. Inspired by this approach, we propose a novel data-driven feature extraction framework that represents facial expression variations as a linear combination of localised basis functions, whose coefficients are proportional to movement intensity. We show that the linear basis functions required by this framework can be obtained by training a sparse linear model with Gabor phase shifts computed from facial videos. The proposed framework addresses generalisation issues that are not addressed by existing learnt representations, and achieves with the *same* learning parameters state-of-the-art results in recognising both posed expressions and spontaneous micro-expressions. This performance is confirmed even when the data used to train the model differ from test data in terms of the intensity of facial movements and frame rate.**

I. Introduction

The recognition of facial expressions from image sequences is fundamental in various applications including social robotics, human-computer interaction and healthcare [1], [2], [3], [4], [5]. Facial expression recognition methods can be dynamic (sequence-based) or static (image-based). While dynamic approaches generally outperform static approaches especially in recognising subtle expressions [5], [6], a key problem for dynamic facial expression analysis is to convert the input sequence into a useful representation. Most approaches use engineered representations, such as Gabor motion energy [7], Local Binary Patterns from Three Orthogonal Planes (LBP-TOP) [8] or Local Phase Quantisation from TOP (LPQ-TOP) [9]. However, representations learnt from data [10], [11], [12], [13] may achieve higher performance without requiring domain expertise [11], [13].

A learnt representation needs to address three important generalisation challenges. First, facial expressions manifested by real-world emotions cause a wide range of movements and therefore the representation should be able to cover a wide *range* of expression intensities, from subtle to pronounced expressions [3]. Micro-expressions are a sub-class of subtle expressions that are characterized by small facial appearance changes and short duration — they can be as short as 1/25 seconds [14]. However, existing learnt (dynamic) representations [10], [13], [12], [11] have not been validated for recognising subtle expressions, even though the recognition of such expressions is an important motivation in using a dynamic representation instead of a simpler static representation [5]. Second, learning a representation usually requires image sequences labelled with expressions and the *applicability* of the features learnt for a particular set of expressions may not extend to the recognition of other expressions. For example, features learnt for the six basic emotions (happiness, sadness, surprise, disgust, fear and anger) [3] may not be useful for recognising other emotion categories, such as contempt or boredom, or when emotions are represented with a continuous affect model [15]. Third, training and test sequences may contain *temporal inconsistencies*, that is, there may be mismatches in terms of frame rate, the speed at which expressions evolve, or the temporal phases contained in the sequences. For example, training sequences may contain all the phases of an expression (*i.e.* neutral, onset, apex, offset [16]), whereas test sequences may contain only some of them (*e.g.* neutral, onset, apex).

In this paper, we propose an unsupervised learning framework that addresses the afore-mentioned generalisation challenges when producing a dynamic representation. Our representation is inspired from FACS (Facial Action Coding System), which is developed by psychologists to analyse expressions for various purposes [16] including the recognition of emotions [16], depression [17] or pain [18]. FACS is similar to a dictionary of elementary facial movements, termed Action Units (AUs), that can be used to represent more complex facial expressions. The AUs describe *localised* movements (*e.g.* AU1 is inner brow raising, AU4 is brow lowering), and each AU is associated with an *intensity* score. These two properties are fundamental for an effective representation: localised movements promote a compact representation, as different facial expressions may contain some common movements (*e.g.* AU1 occurs both in expressions of sadness and fear); and intensity scores enable the usage of the same AU to represent a subtle or an pronounced version of the same facial movement. Furthermore, since our framework is unsupervised, the representation learnt on a specific set of expression labels (*e.g.* pronounced six basic expressions [19]) can be used on a test set with other expression labels (*e.g.* three classes of micro-expressions [20]).

We learn a linear model whose basis functions correspond to localised facial movements and basis coefficients are proportional to movement intensity. The model can therefore

E. Sariyanidi and A. Cavallaro are with the Centre for Intelligent Sensing, Queen Mary University of London, UK. H. Gunes is with the Computer Laboratory, University of Cambridge, UK.
E-mail: {e.sariyanidi, a.cavallaro}@qmul.ac.uk, hatice.gunes@cl.cam.ac.uk
The work of E. Sariyanidi and H. Gunes are partially supported by the EPSRC under its IDEAS Factory Sandpits call on Digital Personhood under Grant EP/L00416X/1.

TABLE I
DYNAMIC FACIAL REPRESENTATIONS IN THE STATE OF THE ART. †REPRESENTATIONS THAT CAN BE TRAINED WITHOUT LABELS, BUT ACHIEVE LOWER PERFORMANCE IN THIS CASE. N/A: NOT APPLICABLE. LBP-TOP: LOCAL BINARY PATTERNS FROM THREE ORTHOGONAL PLANES; DTAGN: DEEP TEMPORAL APPEARANCE-GEOMETRY NETWORK; 3DCNN-DAP: 3D CONVOLUTIONAL NEURAL NETWORK DEFORMABLE ACTION PARTS.

Ref.	Approach	Engineered	Learnt	Static	Dynamic	Needs Training Labels	Adressed Temporal Inconsistencies	Validation by Cross-database Representation Learning	Validated on Pronounced Expressions	Validated on Subtle Expressions
[21]	Non-negative Matrix Factorisation		✓	✓		✓†	N/A		✓	
[22]	Deep Learning		✓	✓		✓	N/A		✓	
[23]	Deep Generative Learning		✓	✓			N/A		✓	
[8]	LBP-TOP	✓			✓	N/A	N/A	N/A	✓	✓
[7]	Gabor Motion Energy	✓			✓	N/A	N/A	N/A	✓	✓
[10]	DTAGN		✓		✓	✓			✓	
[12]	Expressionlets		✓		✓	✓†			✓	
[11]	Spatio-temporal RBM		✓		✓	✓			✓	
[13]	3DCNN-DAP		✓		✓	✓			✓	
Proposed: Facial Bases (F-Bases)			✓		✓		✓	✓	✓	✓

represent facial activity as a (linear) combination of localised movements. Specifically, we show that to learn a linear model where basis coefficients are proportional to movement intensity, we must convert sequences into a representation where monotonic increases in movement intensity correspond to monotonic variations. For this purpose we use Gabor phase shifts [24]. Then, we show that basis functions that correspond to localised facial activity can be learnt by training a sparsity-imposed linear model with Gabor phase shift data computed from facial videos. The proposed model is generative, which enables us to synthesise facial expression sequences and discuss the properties of the learnt bases. Our framework is inspired by developments in human vision research and is similar to that of Cadieu and Olshausen [25] in that it models higher-level structure from the phase and magnitude of (complex) local coefficients. However, their model produces global basis functions rather than localised bases. Global basis functions are suitable for arguing for the existence of motion-sensitive but shape-insensitive representation in the human visual cortex. On the contrary, our localised bases are shape-selective as each basis pertains to a specific facial region.

In summary, the contributions of this paper are as follows. To the best of our knowledge, we propose the first learnt facial expression representation that has been designed for analysing expressions at a range of intensities, and has been validated for recognising both pronounced expressions *and* micro-expressions; and the first learnt dynamic facial expression representation that addresses the temporal inconsistencies that may exist between the training and test sequences. Moreover, we show that learning a sparseness-imposed generative linear model from Gabor phase shifts of facial expression sequences yields basis functions that correspond to localised facial movements.

The paper is organised as follows. Section II reviews existing approaches for learnt facial expression representations. Section III presents the problem formulation. Section IV describes the framework for learning bases of facial activity. Next, Section V presents a qualitative analysis of the learnt bases and describes how they are used for automatic expression recognition. Experimental results are discussed in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORK

In this section we review learnt representations for automatic facial expression recognition. We focus our discussion on the generalisation issues discussed in the previous section, namely (i) the ability to recognise expressions of different intensities, (ii) whether expression labels are required for learning, and (iii) the sensitivity to temporal inconsistencies. We also discuss the difficulties inherent to automatic AU recognition, which is an alternative approach that relates to our work.

The development of learnt facial expression representations started with static representations, which encode the expression in each frame of a sequence independently from neighbouring frames [5]. A variety of approaches are explored, such as non-negative matrix factorisation [21], deep learning [23], [26], [22] or sparse representation [27]. Most of these approaches also learn localised bases [21], [22], [26]. However, static representations are inherently limited in their ability to recognise subtle expressions [6], which are very informative in recognising and modelling real-world affective interactions [5].

Subtle expressions are better recognised when the temporal variation among the frames of a sequence is encoded [6]. Researchers exploited this finding by using (engineered) dynamic representations, such as LBP-TOP [8], [20] or Gabor motion energy [7]. Subsequent studies questioned the optimality of engineered dynamic representations, and aimed to learn representations from video volumes. Liu *et al.* [13] proposed a deep architecture that learns deformable facial parts. Jung *et al.* [10] proposed a deep architecture that comprises two networks – one that learns from facial appearance and another that learns from facial feature points – and showed that the joint learning of the two generally improves performance. Elaiwat *et al.* [11] proposed a restricted Boltzmann machine (RBM) network that, unlike typical deep models, is shallow and therefore easier to optimise. The key feature of this RBM network is to disentangle expression-related image transformations from transformations that are not related to expressions. Liu *et al.* [12] proposed the so-called Expressionlets, which are based on clustering cuboids of pre-defined sizes extracted from facial sequences in order to model the manifold of facial expression variations.

The learnt representations listed so far outperform engineered features in datasets with large (*i.e.* pronounced) and posed facial activity, such as CK+ [28] or MMI [19]. Moreover, those representations are validated only through within-dataset tests, *i.e.* the representations are learnt and tested on the same dataset, where the temporal order of the facial expression phases (*i.e.* neutral-onset-apex for CK+ and neutral-onset-apex-offset-neutral for MMI) and the frame rate of the sequences are the same. Further validation is needed to test whether the learnt features produce meaningful representations on test sequences with different frame rate or order of temporal expression phases compared to the training sequences. In addition to the above, those representations are tested on six basic expressions and use the training labels of sequences during learning. Therefore, their usefulness in other facial expression recognition tasks such as the recognition of arousal-valence labels and micro-expressions also requires further validation. (Note that while Expressionlets can be used without labels, their performance drops considerably when done so [12].) Finally, the long-standing open issue in facial expression recognition is that pipelines trained on pronounced expressions do not generalise to subtle expressions [3], [2], [5] and none of the learnt dynamic representations addressed this issue by validating on sequences with subtle expressions.

Table I summarises the above discussion and compares existing representations. Table I also highlights some advantages of engineered representations. Owing to their simplicity, engineered representations require no labelled sequences and therefore are generic in terms of the final task (*e.g.* micro-expression recognition [29] or six basic expressions [8]). Moreover, inconsistency in frame rate or temporal order of expressions is not an issue for engineered representations as they require no training sequences. Also, engineered representations have been validated for subtle facial expression analysis [29], [7]. It is therefore desirable for learnt representations to possess these advantages.

An alternative approach to facial expression analysis is to develop pipelines that recognise AUs automatically, and to use the output of those pipelines as an intermediate representation for higher-level recognition tasks, such as the recognition of six basic emotions [5] or pain [18]. However, recognising AUs automatically is a challenging problem and each AU requires a dedicated supervised machine learning pipeline. The first challenge is data annotation: AU labelling is a time consuming task as it can take up to 100 minutes to label one minute of video [30]. At least two FACS coders who have undergone a specialised training are required and labels cannot be used without inter-coder agreement, which can be particularly low for low-intensity AUs [31]. To learn different intensities of the same AU, statistical learning algorithms need AU labels across a range of intensities, thus increasing exponentially the need for data. Moreover, even if algorithms could recognise each AU perfectly, it is not guaranteed that a new AU combination will be recognised as AU combinations are not always additive [32]. Discovering useful mappings via statistical learning from annotated data is another challenge, due to data imbalance between positive and negative samples [33] or to generalisation across subjects (*i.e.* identity bias) [5]. As a result, the recognition of AUs, and their intensities in particular, is still an open problem.

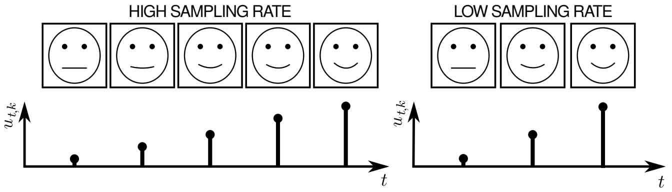


Fig. 1. Illustration that depicts how a basis can provide useful information on videos with different frame rates. Let a basis A_k model the lip corner pulling that occurs during a smile. When a sequence is recorded at a lower rate, the apparent motion speed increases and the expression-related movement occurs at a higher intensity. If the basis coefficient $u_{t,k}$ is proportional to movement intensity as in Eq. (1), then the basis A_k can help recognise the smile independently of whether it is collected at a high or low frame rate. The only difference the frame rate change causes is the rate at which $u_{t,k}$ increases.

III. Problem formulation

Let $\mathbf{I} \in \mathbb{R}^{X \times Y \times T}$ be an image sequence that contains either a whole face or part of a face (*e.g.* the mouth). Let us assume that rigid registration errors have been removed with a registration technique (e.g. [46]). Moreover, let us assume that the motion between two consecutive frames, $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$, is due to facial expression variations only. Let $f(\mathbf{I}_{t-1}, \mathbf{I}_t)$ be a function that represents the motion between $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$ at a local level (*e.g.* an optical flow function) with a D-dimensional vector.

We aim to find a linear transformation that can reconstruct the overall facial activity in terms of local movements. Let $\{A_k\}_{k=1}^{K_A}$ be the set that contains the K_A *basis vectors* of this transformation. Then, we can represent the linear transformation we seek as:

$$f(\mathbf{I}_{t-1}, \mathbf{I}_t) = \sum_{k=1}^{K_A} A_k u_{t,k} + \boldsymbol{\epsilon}_t, \tag{1}$$

where $\boldsymbol{\epsilon}_t$ represents reconstruction error. We want the basis coefficients, $u_{t,k}$, to be proportional to movement intensity. For example, if the basis vector A_k corresponds to an eyebrow raising, then a small (large) $u_{t,k}$ value should mean that the pair $\mathbf{I}_{t-1}, \mathbf{I}_t$ contains a small (large) eyebrow movement.

This linear transformation has two advantages: (i) it enables the separation of subtle and large facial motions through the magnitude of coefficients; and (ii) the bases $\{A_k\}_k^{K_A}$ can be used independently from the video frame rate, as variations in video frame rate (*i.e.* apparent motion speed) cause variation only in the rate at which the coefficients $u_{t,k}$ change over time (Fig. 1).

IV. Bases of Facial Activity

A. The learning framework

Facial expressions increase in their intensity gradually and monotonically until they reach their apex [16]. To capture this aspect via Eq. (1), the magnitudes $|u_{t,k}|$ should vary gradually and monotonically as t increases. The bases A_k are fixed, which implies that for Eq. (1) to hold we must use a motion representation $f(\mathbf{I}_{t-1}, \mathbf{I}_t)$ whose elements are also changing gradually and monotonically as t increases. Therefore, we cannot simply use the difference between the frames (*i.e.* derivative, $\mathbf{I}_t - \mathbf{I}_{t-1}$) as derivatives undergo abrupt changes. One representation can be computing motion vectors via optical flow. However, motion vectors can be erroneous, particularly when representing subtle movements [34] or when computed from untextured regions such as cheeks [35].

To encode local motion without requiring the computation of motion vectors explicitly we chose to infer local motion through Gabor wavelets [24]. A frame $\mathbf{I}_t$ can be recovered from D complex Gabor wavelets $\{W_d\}_{d=1}^{D}$ as [36], [37]:

$$\mathbf{I}_t = \sum_{d=1}^{D} \Re\{z^*_{t,d} W_d\}, \tag{2}$$

where $\Re\{\cdot\}$ is the real part of the argument, $*$ denotes conjugation, and $\mathbf{z}_t = (z_{t,1}, z_{t,2}, \ldots, z_{t,D})$ is the vector of complex Gabor coefficients. Each $z_{t,d}$ can be decomposed into its phase, $\phi_{t,d}$, and magnitude, $\rho_{t,d}$, as:

$$z_{t,d} = \rho_{t,d} e^{j\phi_{t,d}}. \tag{3}$$

Gabor wavelets have limited spatial support. The magnitude $\rho_{t,d}$ and phase $\phi_{t,d}$ take non-zero values when a visual element (*e.g.* an edge) within the wavelet's spatial support causes texture variation. The phase $\phi_{t,d}$ is sensitive to the position of the element and, compared to the magnitude $\rho_{t,d}$, is less sensitive to the intensity of the element (see Fig. 2). Since phase is sensitive to position, the *phase shift*

$$\dot{\phi}_{t,d} = \phi_{t,d} - \phi_{t-1,d} \tag{4}$$

is sensitive to motion [36]. Importantly, phase varies proportionally with the position, as shown in Fig. 2d. Since a Gabor wavelet W_d has local spatial support and is tuned to a specific orientation [37], the phase of one wavelet, $\phi_{t,d}$, can represent motion only locally and has limited ability to represent motion in arbitrary orientations. A complete motion representation can be obtained with a set of wavelets, $\{W_d\}_{d=1}^{D}$, that span the whole image and are tuned to various orientations [37]. Such a representation allows us to encode rigid (*e.g.* global rotations, translations) or non-rigid motions (*e.g.* local rotations, translations) across the image [24].

We can now rephrase our objective as follows. We aim to learn a generative linear model that can represent any expression-induced phase shift pattern $\dot{\boldsymbol{\phi}}_t = (\dot{\phi}_{t,1}, \dot{\phi}_{t,2}, \ldots, \dot{\phi}_{t,D})$ as:

$$\dot{\boldsymbol{\phi}}_t = \sum_{k=1}^{K_A} A_k u_{t,k} + \boldsymbol{\epsilon}^u_t = \mathbf{A}\mathbf{u}_t + \boldsymbol{\epsilon}^u_t. \tag{5}$$

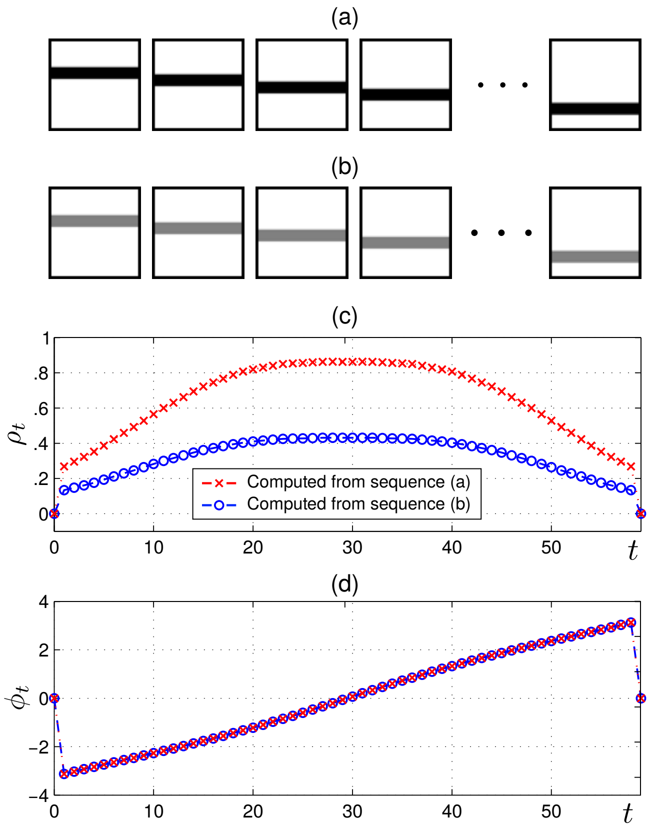


Fig. 2. Illustration that highlights the ability of Gabor phase to encode motion. (a) Exemplar sequence that contains a horizontal bar moving vertically with a constant speed. (b) A sequence that is identical to the one in (a) except that the pixel intensity of the bar is multiplied by 0.5. (c) The magnitude, ρ_t, computed from a Gabor wavelet that is located in the center of the moving images. (d) The phase computed from the same Gabor wavelet. Note that the magnitude changes non-monotonically over time and is sensitive to the intensity of the bar. The phase of the Gabor coefficient, ϕ_t, increases monotonically and is not sensitive to the intensity of the bar.

Note that this equation is a special form of (1). The term $\boldsymbol{\epsilon}^u_t$, which accounts for modelling errors, is assumed to be drawn from a (circular) Normal distribution whose random variables, $\epsilon^u_{t,d}$, are independent from one another and are modelled as $P(\epsilon^u_{t,d}) \propto \exp[\kappa \cos(\epsilon^u_{t,d})]$ where κ is the concentration parameter.

In generative learning, the basis transformation (*i.e.* $\mathbf{A}$) that best describes a given dataset of N i.i.d. samples, $\mathcal{D}_{\dot{\boldsymbol{\phi}}} = \{\dot{\boldsymbol{\phi}}^n\}_{n=1}^{N}$, is the one that maximises the likelihood [38]:

$$\begin{aligned} P(\mathcal{D}_{\dot{\boldsymbol{\phi}}}|\mathbf{A}) &= \prod_{n=1}^{N} P(\dot{\boldsymbol{\phi}}^n|\mathbf{A}) \\ &= \prod_{n=1}^{N} \int P(\dot{\boldsymbol{\phi}}^n|\mathbf{A}, \mathbf{u}) P(\mathbf{u}) d\mathbf{u}. \end{aligned} \tag{6}$$

However, maximising $P(\mathcal{D}_{\dot{\boldsymbol{\phi}}}|\mathbf{A})$ alone may not necessarily yield localised bases.

We guide the maximisation process to learn localised bases by incorporating prior distributions on coefficients $u_{t,k}$ and by imposing constraints on bases A_k. A facial expression generally involves a small proportion of all possible atomic movements that a face can produce. For example, FACS represents any of the six basic expressions with at most 7 out of the 46 AUs [16]. Therefore, only a small proportion of coefficients $u_{t,k}$ must have large values, and the remaining coefficients must be zero or relatively very small. This can be enforced by using a prior distribution on $u_{t,k}$ that favours $u_{t,k}$ being zero with a high and kurtotic peak, such as a zero-mean Cauchy distribution [38]. Also, the prior should favour small differences in $u_{t,k} - u_{t-1,k}$ as expressions evolve gradually over time. This can be incorporated with a Gaussian distribution centred on $u_{t,k} - u_{t-1,k}$ [39]. Then the overall prior on $u_{t,k}$ becomes:

$$P\left(u_t|u_{t-1}\right) = \frac{1}{Z_u} e^{-\lambda_u \log(1+u_t)} e^{-\beta_u (u_t - u_{t-1})^2}, \tag{7}$$

where λ_u and β_u are the scale and precision parameters of the Cauchy and Gaussian distribution, respectively, and Z_u is the normalisation coefficient ensuring that the distribution sums to 1. Note that the subscript k is dropped for clarity.

For a basis A_k to be localised, most of its elements must be zero, and non-zero elements should pertain to spatially nearby regions. Hoyer [40] proposed a technique to produce such localised bases by enforcing the following sparseness metric:

$$\mathscr{S}(A_k) = \frac{\sqrt{D} - \frac{||A_k||_1}{||A_k||_2}}{\sqrt{D} - 1}, \tag{8}$$

where $||\cdot||_1$ and $||\cdot||_2$ denote the L_1 and L_2 norms, respectively. The sparser A_k, the higher the $\mathscr{S}(A_k)$. Sparse and localised bases are obtained by pre-defining a sparseness rate S_A and enforcing all bases to follow this rate (*i.e.* $\mathscr{S}(A_k) = S_A$) during optimisation (see Section IV-C).

B. Static vs. Dynamic Bases

When there is no expression variation in a sequence, there is no motion and the phase shifts $\dot{\boldsymbol{\phi}}_t$ become zero. The model must therefore be capable of analysing the expression from the *facial configuration*; that is, the appearance variation that has already been generated by the expression (Fig. 3). This can be achieved by learning *static bases*, in a similar fashion to learning dynamic bases. Dynamic bases were learnt from phase shifts $\dot{\boldsymbol{\phi}}_t$, whereas static bases are learnt from magnitudes:

$$\boldsymbol{\rho}_t = (\rho_{t,1}, \rho_{t,2}, \ldots, \rho_{t,D}), \tag{9}$$

which relate to the persistent structure in images [25]. While a dynamic basis pertains to a localised facial *movement* (*e.g.* raising an eyebrow), a static basis describes a particular facial *configuration* localised in space (*e.g.* a raised eyebrow).

We seek to learn a generative linear model that can represent a magnitude pattern, $\boldsymbol{\rho}_t$, generated by any facial configuration. Specifically, we use the log-magnitudes, as taking logarithm linearises the dependencies between magnitudes [25]:

$$\log \boldsymbol{\rho}_t = \sum_{k=1}^{K_B} B_k v_{t,k} + \boldsymbol{\epsilon}_t^v = \mathbf{B}\mathbf{v}_t + \boldsymbol{\epsilon}_t^v, \tag{10}$$

where $\{B_k\}_{k=1}^{K_B}$ are the static bases, $v_{t,k}$ are the static coefficients and $\boldsymbol{\epsilon}_t^v$ is a noise term that is drawn from a Normal distribution, *i.e.* $p(\boldsymbol{\epsilon}_t^v) \sim \mathcal{N}(0, \sigma_\rho)$. During learning, we impose priors and constraints similar to the dynamic bases. We assume that $\log \boldsymbol{\rho}_t$ can be recovered sparsely, that is, using a small proportion of bases, and that the facial appearance changes gradually over time.

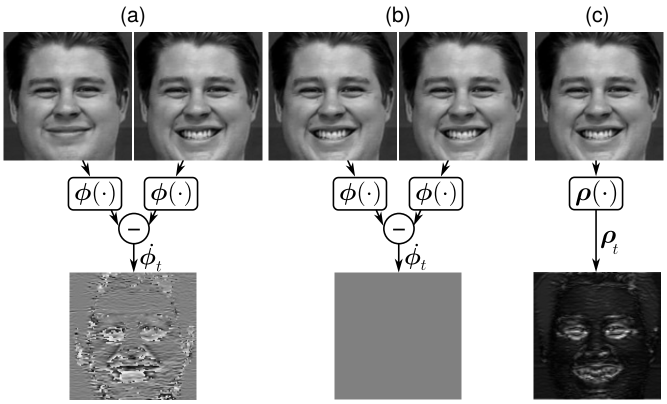


Fig. 3. Example of the importance of the magnitude to recognize an expression. For clarity, magnitude and phase responses are illustrated only for one Gabor filter. (a) The phase shift provides useful information when there exist expression variations between consecutive frames. (b) The phase shifts are not informative in the absence of expression variations. (c) The magnitude computed from a (static) frame provides useful information to recognise the expression in the absence of expression variations.

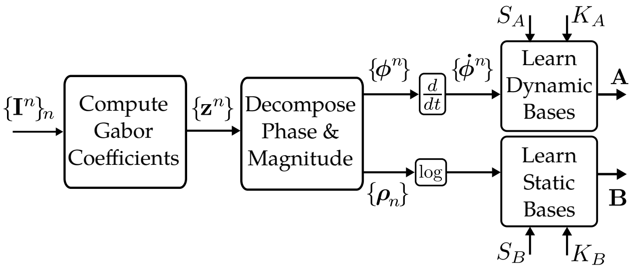


Fig. 4. Illustration of how bases are learnt from a dataset, $\mathcal{D} = \{\mathbf{I}^n\}_{n=1}^N$. The subscript n is dropped in later stages for clarity. The depicted variables are listed in Table II along with their dimensionality.

The resulting prior $P\left(v_{t,k}|v_{t-1,k}\right)$ is identical to Eq. (7) in form but differs in its parameters; the scale of the Cauchy distribution, the precision of the Gaussian and the normalisation coefficient are denoted respectively with λ_v, β_v and Z_v. The overall pipeline of the proposed model is illustrated in Fig. 4. The variables referred to in Fig. 4 are listed in Table II along with their dimensionality.

C. Optimisation

We can formulate the learning of static and dynamic bases as the following optimisation problem.

Problem 1: Given a dataset of phase shifts, $\mathcal{D}_\phi = \{\dot{\boldsymbol{\phi}}^n\}_{n=1}^N$, a dataset of magnitudes, $\mathcal{D}_\rho = \{\boldsymbol{\rho}^n\}_{n=1}^N$, the number of dynamic and static bases, K_A, K_B, and sparseness ratios S_A, S_B, find $\mathbf{A}^* \in \mathbb{R}^{D \times K_A}$ and $\mathbf{B}^* \in \mathbb{R}^{D \times K_B}$ that satisfy:

$$\mathbf{A}^* = \arg\max_{\mathbf{A}} \left[\log P(\mathcal{D}_{\dot{\phi}}|\mathbf{A})\right], \tag{11}$$

$$\mathbf{B}^* = \arg\max_{\mathbf{B}} \left[\log P(\mathcal{D}_\rho|\mathbf{B})\right], \tag{12}$$

TABLE II
LIST OF VARIABLES WITH THEIR SYMBOLS AND THEIR DIMENSIONS.

$\mathbf{I} \in \mathbb{R}^{X \times Y \times T}$	An image sequence
$\mathbf{I}_t \in \mathbb{R}^{X \times Y}$	t^{th} frame of $\mathbf{I}$
$\mathcal{D} = \{\mathbf{I}^n\}_{n=1}^N$	Dataset to learn the bases from
$\mathbf{z} \in \mathbb{C}^{D \times T}$	Gabor coefficients of an $\mathbf{I}$
$\dot{\boldsymbol{\phi}} \in \mathbb{R}^{D \times (T-1)}$	Phase shifts computed from $\mathbf{z}$
$\boldsymbol{\rho} \in \mathbb{R}^{D \times T}$	Magnitudes computed from $\mathbf{z}$
$\dot{\boldsymbol{\phi}}_t, \boldsymbol{\rho}_t \in \mathbb{R}^D$	Phase shift, magnitude for t^{th} frame
$\mathbf{A} \in \mathbb{R}^{D \times K_A}$	Dynamic basis transformation matrix
$A_k \in \mathbb{R}^D$	k^{th} dynamic basis (*i.e.* k^{th} column of $\mathbf{A}$)
$S_A \in \mathbb{R}[0,1]$	Sparseness ratio of bases A_k
$\mathbf{u} \in \mathbb{R}^{K_A \times (T-1)}$	Dynamic basis coefficients of $\mathbf{I}$
$\mathbf{u}_t \in \mathbb{R}^{K_A}$	Dynamic basis coefficients of $\mathbf{I}_t$
$\mathbf{B} \in \mathbb{R}^{D \times K_B}$	Static basis transformation matrix
$B_k \in \mathbb{R}^D$	k^{th} static basis (*i.e.* k^{th} column of $\mathbf{B}$)
$S_B \in \mathbb{R}[0,1]$	Sparseness ratio of bases B_k
$\mathbf{v} \in \mathbb{R}^{K_B \times T}$	Static basis coefficients of $\mathbf{I}$
$\mathbf{v}_t \in \mathbb{R}^{K_B}$	Static basis coefficients of $\mathbf{I}_t$

Algorithm 1 Learn dynamic bases

Input: Dataset of facial videos $\mathcal{D} = \{\mathbf{I}^n\}_{n=1}^N$, $\tau_A^{\max}$, $\tau_u^{\max}$
Output: Dynamic basis transformation $\mathbf{A} \in \mathbb{R}^{D \times K_A}$

```
Compute Gabor coefficients $\mathcal{D}_z = \{\mathbf{z}^n\}_n$ from $\mathcal{D}$
Compute phases $\mathcal{D}_\phi = \{\boldsymbol{\phi}^n\}_n$ from $\mathcal{D}_z$
Compute phase shifts from $\mathcal{D}_{\dot{\phi}} = \{\dot{\boldsymbol{\phi}}^n\}_n$ from $\mathcal{D}_\phi$
Initialise $A_k$ with random values $\forall k \in \{1, 2, \dots, K_A\}$
Initialise $\mathbf{u}^n$ with random values $\forall n \in \{1, 2, \dots, N\}$
$\tau_A \leftarrow 0$
repeat
    for each sample $\dot{\boldsymbol{\phi}}^n$ do
        $\tau_u \leftarrow 0$
        repeat
            $\mathbf{u}^n \leftarrow \mathbf{u}^n + \alpha_u^{(\tau_u)} \Delta \mathbf{u}^n$
            $\tau_u \leftarrow \tau_u + 1$
        until $\tau_u^{\max}$ is reached
    end for
    for each $A_k$ do
        $A_k \leftarrow A_k + \alpha_A^{(\tau_A)}$
        $A_k \leftarrow \text{project}(A_k; S_A)$
    end for
    $\tau_A \leftarrow \tau_A + 1$
until $\tau_A^{\max}$ is reached
```

under the constraints

$$\mathscr{S}(A_k) = S_A, \ \forall k \in \{1, 2, \dots, K_A\}, \tag{13}$$

$$\mathscr{S}(B_k) = S_B, \ \forall k \in \{1, 2, \dots, K_B\}. \tag{14}$$

Maximising the likelihoods in Eq. (11–12) is equivalent to minimising the negative log-likelihoods, denoted as $E_\phi = -\log P(\mathcal{D}_{\dot{\phi}}|\mathbf{A})$ and $E_\rho = -\log P(\mathcal{D}_\rho|\mathbf{B})$.

To minimise E_ϕ and E_ρ, we need their closed-form expressions, which are intractable due to the integrals such as those in (6). We simplify the integrals by assuming that the integrands are highly peaked around the coefficients $\mathbf{u}$ (or $\mathbf{v}$) that maximise the integrands, and by replacing the integrals with the maximal value of their integrands [38]. Then, using the fact that $\epsilon_{t,d}^u, \epsilon_{t,d}^v$ are generated from circular Normal and Normal distributions, and using the priors $P(u_t|u_{t-1})$ and $P(v_t|v_{t-1})$, we can approximate E_ρ, E_ϕ as [25]:

$$E_\phi \approx \sum_{n=1}^N \sum_{t=2}^T \sum_{d=1}^D \Big[\kappa \cos(\dot{\phi}_{t,d}^n - [\mathbf{A}\mathbf{u}_t^n]_d) + \lambda_u \log(1 + u_{t,d}^n) + \beta_u (u_{t,d}^n - u_{t-1,d}^n)^2\Big], \tag{15}$$

$$E_\rho \approx \sum_{n=1}^N \sum_{t=1}^T \sum_{d=1}^D \Big[\frac{1}{\sigma_\rho^2} (\log \rho_{t,d}^n - [\mathbf{B}\mathbf{v}_t^n]_d)^2 + \lambda_v \log(1 + v_{t,d}^n) + \beta_v (v_{t,d}^n - v_{t-1,d}^n)^2\Big], \tag{16}$$

where $[\cdot]_i$ indicates the i^{th} element of its (vector) argument.

Since the approximations above use only the $\mathbf{u}, \mathbf{v}$ values that maximise the integrands in Eq. (6), we must follow a two-fold optimisation scheme [38]: First, fix $\mathbf{A}$ (or $\mathbf{B}$) and minimise *w.r.t.* $\mathbf{u}$ (or $\mathbf{v}$), and then vice versa. This two-fold minimisation is carried out until a maximal number of iterations $\tau_A^{\max}$ (or $\tau_B^{\max}$) is reached. This minimisation requires the gradients of Eq. (15–16) with respect to the basis functions, $\Delta A_{dk}, \Delta B_{dk}$, and with respect to the coefficients, $\Delta u_{t,k}^n, \Delta v_{t,k}^n$. The former are (up to constant divisive factors):

$$\Delta A_{dk} = \kappa \sum_{n=1}^N \sum_{t=2}^T \sin(\dot{\phi}_{t,d}^n - [\mathbf{A}\mathbf{u}_t^n]_d) u_{t,k}^n, \tag{17}$$

$$\Delta B_{dk} = \sum_{n=1}^N \sum_{t=1}^T \frac{2}{\sigma_\rho^2} (\log \rho_{t,d}^n - [\mathbf{B}\mathbf{v}_t^n]_d) v_{t,k}^n. \tag{18}$$

The gradients with respect to the coefficients are (up to constant divisive factors):

$$\Delta u_{t,k}^n = \kappa \sum_{d=1}^D (\sin \dot{\phi}_{t,d}^n - [\mathbf{A}\mathbf{u}_t^n]_d) A_{dk} - \lambda_u \frac{1}{2 + 2(u_{t,k}^n)^2} - 2\beta_u (u_{t,k}^n - u_{t-1,k}^n), \tag{19}$$

$$\Delta v_{t,k}^n = \frac{2}{\sigma_\rho^2} \sum_{d=1}^D (\log \rho_{t,d}^n - [\mathbf{B}\mathbf{v}_t^n]_d) B_{dk} - \lambda_v \frac{1}{2 + 2(v_{t,k}^n)^2} - 2\beta_v (v_{t,k}^n - v_{t-1,k}^n). \tag{20}$$

Using these gradients, we compute $\mathbf{u}_t$, $\mathbf{A}$ by updating them iteratively, where the update rules for an iteration τ are:

$$u_{t,k}^n \leftarrow u_{t,k}^n + \alpha_u^{(\tau)} \Delta u_{t,k}^n, \tag{21}$$

$$A_{dk} \leftarrow A_{dk} + \alpha_A^{(\tau)} \Delta A_{dk}, \tag{22}$$

where $\alpha_u^{(\tau)}, \alpha_A^{(\tau)}$ are the *learning rates* for iteration τ. (Similar update rules are defined for $\mathbf{v}_t, \mathbf{B}$.) While the learning rates can simply be set to fixed values, this may cause very slow convergence [41]. Efficient algorithms use learning rates defined automatically at each update step [41]. To this end, we use the Barzilai-Borwein method [42] for estimating $\alpha_u^{(\tau)}$

and adaptive steepest descent for estimating $\alpha_A^{(\tau)}$. We use the two respective algorithms while also computing the learning rate for static bases, $\alpha_B^{(\tau)}$, and the learning rate for static coefficients, $\alpha_v^{(\tau)}$.

The constraints in Eq. (13–14) can be satisfied with a number of $L1$ regularisation algorithms [43]. We use the projection algorithm proposed by Hoyer [40] as it has already proved successful in creating localised bases for facial data. While the algorithm was originally used to create bases in the space of (static) raw pixels, we can report that it also creates localised dynamic and static bases that live in the space of Gabor phase shifts and magnitudes, respectively. We denote this projection algorithm as $\text{project}(\cdot)$ and use it to update A_k, B_k in order to satisfy Eq. (13–14) as:

$$A_k \leftarrow \text{project}(A_k; S_A), \tag{23}$$

$$B_k \leftarrow \text{project}(B_k; S_B). \tag{24}$$

Note that this projection algorithm is originally developed for non-negative matrix factorisation. To obtain a basis function with negative values too, we keep the signs of a basis function's values before sparsifying the basis, and then apply these signs to the sparsified basis [40].

Once the basis transformations $\mathbf{A}, \mathbf{B}$ are learnt, we compute the coefficients $\mathbf{u}, \mathbf{v}$ for a new sequence $\mathbf{I}$ as follows. First we compute the sequence's Gabor coefficients, $\mathbf{z}$. Then, we compute $\dot{\boldsymbol{\phi}}$ and $\log \boldsymbol{\rho}$ from $\mathbf{z}$. To obtain $\mathbf{u}$, we initialise $\mathbf{u}$ with random values as in step 5 in Algorithm 1, and finally compute $\mathbf{u}$ iteratively as in steps 8–11 of Algorithm 1. A similar procedure follows for computing $\mathbf{v}$.

V. Synthesis and Analysis with Bases

In this section we visualise the bases (*i.e.* synthesis) and discuss how to use them for automatic facial expression recognition (*i.e.* analysis).

A. Visualisation of Learnt Bases

An advantage of a generative framework is its ability to synthesise sequences. This ability is useful for visualising and interpreting the information encoded in the bases. To visualise a basis A_k, we first select a facial image, $\mathbf{I}_k^0$, and then synthesise frames that reflect the movement encoded in A_k. Using Eq. (2–3), we can represent $\mathbf{I}_k^0$ as:

$$\mathbf{I}_k^0 = \sum_{d=1}^{D} \Re\{\rho_{k,d}^0 e^{-\phi_{k,d}^0} W_d\}. \tag{25}$$

Synthesising an image amounts to altering the phase pattern, $\boldsymbol{\phi}_k^0$, using phase shifts generated through Eq. (5) as:

$$\hat{\mathbf{I}}_k^0(u) = \sum_{d=1}^{D} \Re\{\rho_{k,d}^0 e^{-j(\phi_{k,d}^0 + [A_k u]_d)} W_d\}. \tag{26}$$

We can also synthesise a sequence that visualises a *combination* of bases, for example, a pair of bases as:

$$\hat{\mathbf{I}}_{k+i}^0(u) = \sum_{d=1}^{D} \Re\{\rho_{k,d}^0 e^{-j(\phi_{k,d}^0 + [A_k u]_d + [A_i u]_d)} W_d\}. \tag{27}$$

For representative purposes, we visualise bases learnt from the MMI dataset [19], which contains facial actions with their entire temporal evolution. We set the number of bases to $K_A = 60$ (see Section VI-D for a discussion on the choice of the number of bases for automatic facial expression recognition experiments). To test whether the bases learnt on one dataset enable meaningful inference on *another dataset*, we choose the frames that are used for synthesising, $\mathbf{I}_k^0$, from the CK+ dataset [28]. We learn separate sets of bases for the left eye, right eye and mouth, rather than learning one set of bases for the whole face. The main advantage of this part-based representation is to reduce the temporal texture variation caused by out-of-plane head variations [5] that may interfere with the modelling of facial activity.

Let us consider for example the bases for the left eye and the bases for the mouth. With $K_A = 60$ bases per part, the total number of bases is 120. Let A_{1-60} denote the bases for the left eye and A_{61-120} denote the bases for the mouth.

Fig. 5 visualises the bases learnt by the proposed model. We synthesise three images with three coefficients, $u, 2u$ and $3u$. To highlight where the movement occurs, we show the difference between consecutive frames, *i.e.* $\hat{\mathbf{I}}_k^0(2u) - \hat{\mathbf{I}}_k^0(u)$ and $\hat{\mathbf{I}}_k^0(3u) - \hat{\mathbf{I}}_k^0(2u)$. The difference images show that movement occurs only in a limited spatial region (*i.e.* bases are localised). Furthermore, localised bases are additive in terms of appearance; that is, when a combination of bases is visualised, the appearance variation caused by each basis is identical to that caused by basis alone, given that the bases in the combination are not overlapping spatially. Examples of combinations of bases are illustrated in the two bottom rows of Fig. 5.

It is interesting to notice similarities between some AUs of FACS and the bases A_k shown in Fig. 5. For example, the bases A_{11}, A_{13}, A_{110}, A_{116} resemble the onset phases of AU 45 (blink), AU 1+2 (inner, outer brow raiser), AU 11 (nasolabial deepener) and the lip corner pulling that occurs with AU 6+12+25 (cheek raiser, lip corner puller, lips part), respectively. We illustrate more bases in supplementary material[1].

Fig. 5 also highlights correlations among bases, which correspond to redundancy in the information provided by some bases. For example, A_{11} and A_{19} represent a similar eyelid movement. Such correlations are due to person-specific differences in the location of the facial features (*e.g.* eyebrow) or the fact that different bases are modelling different fragments of the same movement (*e.g.* one basis models the onset of a movement while another models a later phase). Nearly half of the bases are not directly linked to a specific facial region or location (*i.e.* are not localised, see Fig. 6). Note that learning a generative model aims at reconstructing training samples and *non-localised* bases may be employed by the generative model to produce the residuals that are needed for the reconstruction of some training samples. In other words, non-localised bases may facilitate the creation of localised bases by producing the residuals that cannot be captured efficiently with localised bases.

Fig. 7 illustrates the variations of the coefficients $u_{t,k}$ for

[1] Please see ftp://spit.eecs.qmul.ac.uk/pub/es/supp.zip.

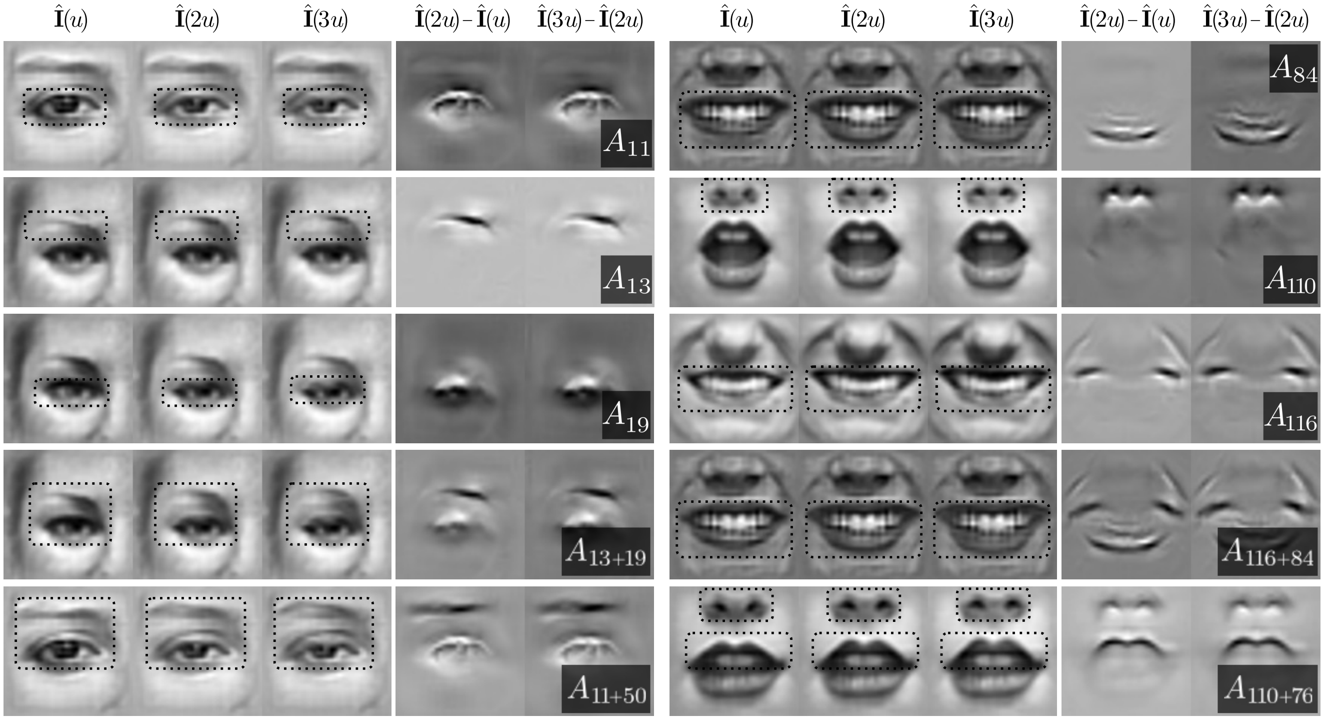


Fig. 5. Illustration of the movement encoded in some of the dynamic bases. To illustrate a basis, A_k, or a combination of bases, A_{k+i}, we synthesise three images with three coefficients: $\hat{\mathbf{I}}^0_k(u)$, $\hat{\mathbf{I}}^0_k(2u)$ and $\hat{\mathbf{I}}^0_k(3u)$. (Note that we drop the subscript k and superscript 0 for clarity.) We encircle the regions with facial movements, and provide the difference images of consecutive frames that also highlight those regions.

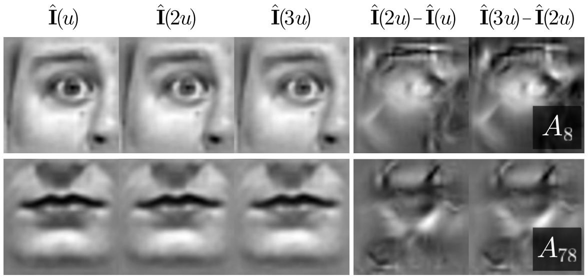

$\hat{\mathbf{I}}(u)$
$\hat{\mathbf{I}}(2u)$
$\hat{\mathbf{I}}(3u)$
$\hat{\mathbf{I}}(2u) - \hat{\mathbf{I}}(u)$
$\hat{\mathbf{I}}(3u) - \hat{\mathbf{I}}(2u)$
A_8
A_{78}

Fig. 6. Sample bases that model non-localised texture variations.

sequences of the MMI dataset [19]. In each sequence we ignore for clarity the initial frames (where there is no facial activity) and the frames after the apex. Moreover, we illustrate only the coefficients produced with the four most activated bases A_k only. Most of the remaining coefficients are very small — the ratio of coefficients that are smaller than 0.1 are 90.2%, 86.5%, 86.3%, and 84.6% for the sequences in Fig. 7a,b,c,d, respectively. The coefficients have small values in the first frames when the expressions are subtle and then get larger as the expressions reach their apex.

B. Automatic facial expression recognition

The learnt bases can be used to extract features for recognising the facial expression in a sequence $\mathbf{I}$. The features can be used as input to a multi-class classifier trained from a set of sequences, $\{\mathbf{I}^n\}_{n=1}^N$ (see Fig. 8).

The first step is computing the static, $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_T)$, and dynamic, $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_T)$, basis coefficients for all frames of $\mathbf{I}$, as described in the last paragraph of Section IV-C. Since the facial expression in $\mathbf{I}$ may not be temporally aligned with the training sequences $\{\mathbf{I}^n\}_{n=1}^N$, we do not use those coefficients directly as features. Instead, we extract features by applying temporal pooling to introduce tolerance against delays or other sources of temporal inconsistencies among test and training sequences.

To extract features from dynamic coefficients $\mathbf{u}$, we first split the coefficients into T_A slices over time, $(\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^{T_A})$, where each slice $\mathbf{u}^\tau$ is a set that contains $Q_A = \lceil \frac{T}{T_A} \rceil$ coefficient vectors, *i.e.*:

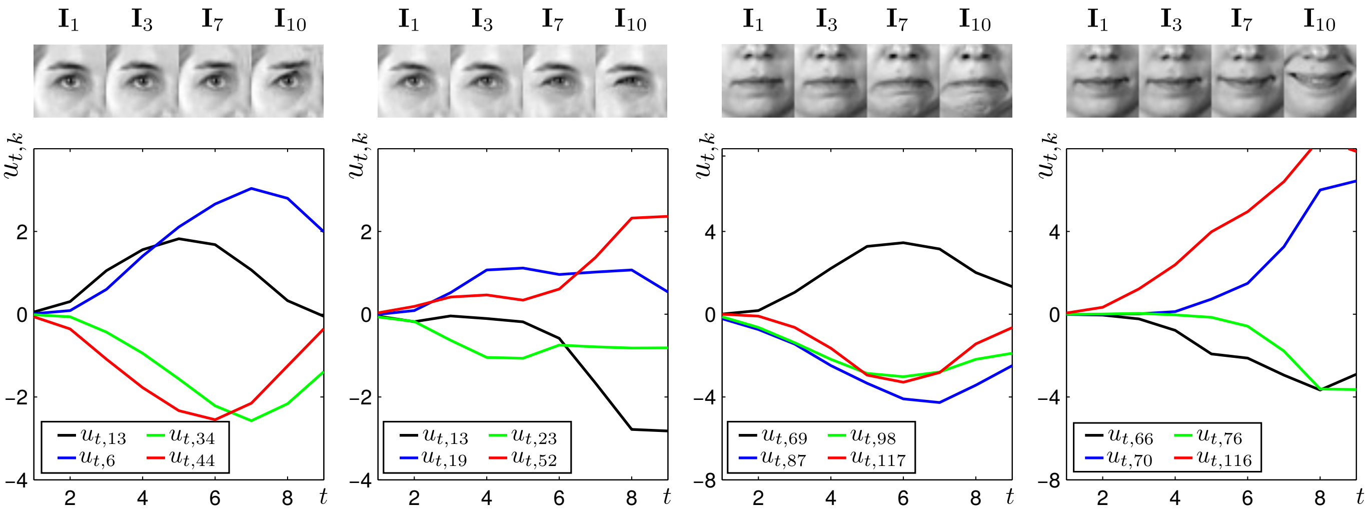
$$\mathbf{u}^\tau = \{\mathbf{u}_{(\tau-1)Q_A+1}, \mathbf{u}_{(\tau-1)Q_A+2}, \dots, \mathbf{u}_{\tau Q_A}\}. \tag{28}$$

Then, we compute histograms for each $\mathbf{u}^\tau$. Specifically, we compute a histogram of H_A bins per basis k such that:

$$\mathbf{h}^{\tau,k} = \text{hist}(\{u_{t',k} : u_{t',k} = [\mathbf{u}_{t'}]_k, \forall \mathbf{u}_{t'} \in \mathbf{u}^\tau\}), \tag{29}$$

where hist$(\cdot)$ is the operator that computes the histogram of its input set. We use histogram pooling as it outperformed simpler approaches (*e.g.* mean, max or standard deviation pooling) in our experiments. We concatenate the histograms computed for all $\tau = 1, 2, \dots, T_A$ and $k = 1, 2, \dots, K_A$. The length of the concatenated histograms is $H_A \times K_A \times T_A$.

We extract features from static coefficients $\mathbf{v}$ in a similar manner, by splitting $\mathbf{v}$ into T_B slices over time, $(\mathbf{v}^1, \mathbf{v}^2, \dots, \mathbf{v}^{T_B})$. However, in this case we use mean and standard deviation pooling, which have lower dimensionality than histogram pooling and generally achieved comparable performance to histogram pooling in our experiments. Specifically, we compute the mean and standard deviation on each of



(a) Left eye sequence, fear expression

(b) Left eye sequence, happiness expression

(c) Mouth sequence, fear expression

(d) Mouth sequence, happiness expression

Fig. 7. The dynamic coefficients $u_{t,k}$ computed on four exemplar sequences that depict two facial parts (left eye, mouth) for two different expressions (fear, happiness). Note that the coefficients are small in the early (*i.e.* subtle) stages of the expression, and they grow larger as the expressions reach their apex.

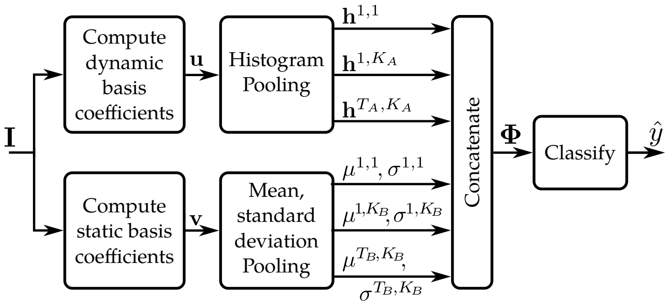

$\mathbf{I}$
Compute dynamic basis coefficients
$\mathbf{u}$
Histogram Pooling
$\mathbf{h}^{1,1}$
$\mathbf{h}^{1,K_A}$
$\mathbf{h}^{T_A,K_A}$
Compute static basis coefficients
$\mathbf{v}$
Mean, standard deviation Pooling
$\mu^{1,1}, \sigma^{1,1}$
$\mu^{1,K_B}, \sigma^{1,K_B}$
$\mu^{T_B,K_B}, \sigma^{T_B,K_B}$
Concatenate
$\mathbf{\Phi}$
Classify
$\hat{y}$

Fig. 8. Block diagram of the proposed end-to-end process to predict $\hat{y}$, the expression in a sequence $\mathbf{I}$, with a pre-trained classifier.

TABLE III

DATASETS USED FOR VALIDATION AND THEIR PROPERTIES. NE: NEUTRAL, ON: ONSET, AP: APEX, OF: OFFSET.

Dataset	CK+	MMI	SMIC
Frame Rate (fps)	12	25	100
Temporal Phases	Ne-On-Ap	Ne-On-Ap-Of-Ne	Mixed
Expression Intensity	Pronounced	Pronounced	Micro-expression
Expression Classes	Six-basic emotions +contempt	Six-basic emotions	Surprise, Positive, Negative

the sets $\mathbf{v}^{\tau}$ for each basis k. We denote the output of these two pooling operators as $\mu^{\tau,k}$ and $\sigma^{\tau,k}$, respectively. The vector of the static features is obtained by concatenating the pooling output for all $\tau = 1, 2, \ldots, T_B$ and $k = 1, 2, \ldots, K_A$. The length of this vector is $2 \times K_B \times T_B$.

Finally, $\mathbf{\Phi}$, the feature vector of $\mathbf{I}$, is obtained by concatenating the pooling output of the dynamic coefficients and the static coefficients. The performance of the proposed facial expression classification process is validated in the next section.

VI. EXPERIMENTS AND RESULTS

To validate the proposed representation, we test its generalisation ability with the recognition of two extreme situations, namely pronounced expressions and micro-expressions. We also evaluate the ability of the learnt bases to recognise facial expressions and to generalise across tasks and databases with different frame rates.

A. Datasets

We validate the generalisation ability of the learnt bases on the Cohn-Kanade (CK+) dataset, the MMI dataset and the SMIC micro-expression dataset, which differ in frame rate, temporal phases of the facial expressions, and magnitude of the expressions (see Table III and Fig. 9).

The CK+ dataset [28] is useful to rank a technique compared to the state of the art as many facial expression recognition systems are evaluated on this dataset. CK+ includes the six basic emotions (anger, disgust, fear, happiness, sadness, surprise) and a non-basic emotion (contempt). We follow the standard protocol of the dataset, *i.e.* leave-one-subject-out (LOSO) cross validation [28]. We use 327 sequences of 118 subjects, *i.e.* all emotion-labelled sequences. The sequences start with a neutral expression and finish at the apex. The MMI dataset [19] is commonly used for the recognition of the six basic emotions. The sequences contain all phases of facial expressions (*i.e.* neutral-onset-apex-offset), and the apex frame is unknown. We use all frontal sequences that are labelled with an emotion. 205 sequences from 31 subjects fit these criteria. We also perform LOSO cross validation. The SMIC micro-expression dataset [20] is useful to evaluate a model's ability to recognise subtle expressions. There are two tests: *micro-expression detection*, which aims to identify whether or not a micro-expression exists in a given sequence, and *micro-expression recognition*, which aims to classify the micro-expression in a sequence as positive, negative or surprise (3-class problem) [20]. We use

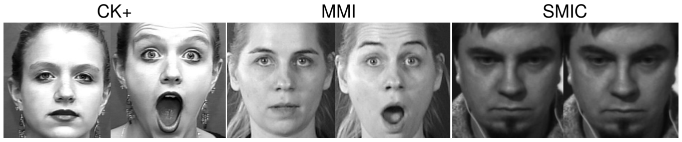


Fig. 9. Examples from the CK+, MMI and SMIC datasets with a neutral frame and a frame with surprise expression, depicting that an emotion can be shown with expressions of different intensities. In the rightmost example, surprise is manifested with a subtle expression that involves an eyebrow movement.

the data collected with a high-speed (100 fps) camera: 164 sequences with micro-expressions and 164 sequences with no micro-expressions.

B. Protocols

We use a C-SVM classifier with linear kernel [44] for CK+ and MMI tests by fixing the C parameter to 10^3 with no further optimisation. The baseline method in SMIC [20] uses a polynomial-kernel SVM, and we also use this kernel when testing on SMIC. We use the same kernel parameters, and learn the C parameter on SMIC with cross-database validation.

As the *evaluation* metric we use the *classification accuracy* in all tests:

$$\alpha = \frac{|\{y^n : y^n = \hat{y}^n\}_{n=1}^{N}|}{N}, \qquad (30)$$

where $|\cdot|$ denotes set cardinality, N is the number of test sequences, and y^n, $\hat{y}^n$ are respectively the ground truth and predictions for the n^{th} sequence.

C. Implementation Details and Computation Time

We learn a part-based representation to reduce the effect of out-of-plane head pose variations, as discussed in Section V-A. We first crop the left eye, the right eye and the mouth components in each frame of a sequence after localising the center of each component with the SDM technique[2] [45]. We crop each component as a square and avoid overlap among different components. The edge size of squares, relative to the inter-ocular distance, δ_{iod}, is set to $1.9\delta_{\text{iod}}$ (as Fig. 10a suggests, smaller squares may reduce performance on CK+ and MMI). Then we register temporally the cropped sequences with the technique in [46], which achieves sub-pixel accuracy. Finally, we re-scale the patches in the registered sequences to 32×32 pixels. As Fig. 10b shows, 32×32 achieves a good balance among the CK+, MMI and SMIC datasets. .

We use the Gabor wavelet set in [47] with 4 orientations and 5 scales, which yields $D = 4468$ Gabor wavelets for frames of size 32×32. We use 4 orientations, instead of the more commonly used 8 [37], to reduce the dimensionality D, as the reconstruction performance with 4 and 8 orientations is similar on our images (see Fig. 11). The noise and prior parameters needed in Eq. (17–20) are set as $\lambda_{v_s}, \lambda_{u_s} = 10; \kappa = 4; \sigma_\rho = 0.25; \lambda_v, \lambda_u = 0.2$. We set $\lambda_{v_s}, \lambda_{u_s}, \kappa$ and σ_ρ based on previous research [25], and λ_v, λ_u based on experiments after noticing little sensitivity to them within the range of $0.1 - 0.4$. We limit maximum iterations as $\tau_v^{\max}, \tau_u^{\max} = 1000$ and $\tau_A^{\max}, \tau_B^{\max} = 250$, which are generally sufficient for convergence. S_A and S_B, are set to 0.75, and we observed qualitatively similar results for the range of $0.65 - 0.85$. We learn separate linear models for each facial part (*i.e.* left eye, right eye and mouth). The learning parameter that has the most significant effect on performance is the number of bases, K_A, K_B. For simplicity, we always set those two quantities to be the same (*i.e.* $K_A = K_B$) rather than optimising them separately. We perform experiments for various values of K_A and analyse its effect in our discussion. For gradient descent optimisation we use [48] and for the $\text{project}(\cdot)$ algorithm we use [40].

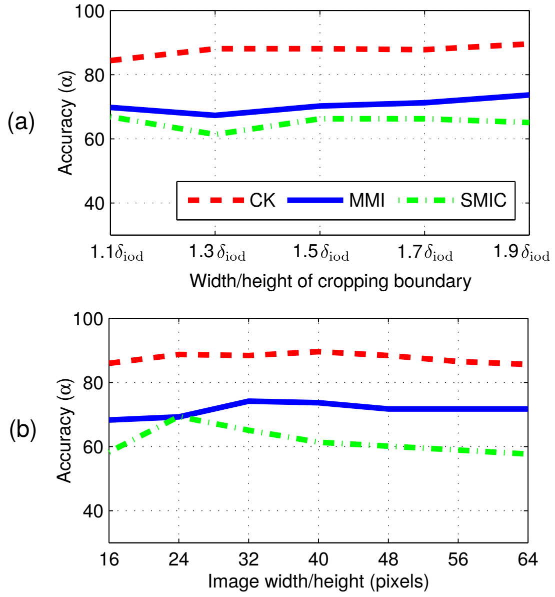


Fig. 10. Performance variation with the dynamic bases with respect to (a) the size of the cropping rectangle in terms of inter-ocular distance, δ_{iod}, and (b) the size of the cropped patches after re-scaling ($K_A = 60$ for the MMI dataset and $K_A = 100$ for the CK+ and SMIC datasets).

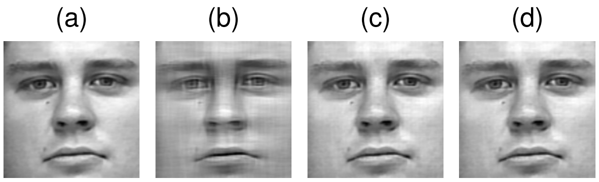


Fig. 11. Reconstruction performance of sets that contain wavelets at 2, 4 and 8 different orientations. (a) Facial image from the CK+ dataset. (b), (c), and (d) show the reconstruction performance of wavelet sets that contain wavelets at 2, 4 and 8 orientations, respectively. Note that when increasing the number of orientations from 2 to 4 there is a significant improvement in reconstruction quality, whereas there is little improvement when increasing the number of orientations from 4 to 8.

We process all sequences to have the same length of frames. In CK+, where the sequences end with the apex of the expressions, we use the last 8 frames as all sequences have at least 8 frames. In MMI and SMIC, the apex of the expressions is unknown and we use all frames; for those datasets, we resize training sequences via temporal interpolation (similarly to [20]) to 10 frames when learning the bases. Temporal interpolation effectively changes the frame rate of the sequences.

[2] The SDM technique provides the corners of the left eye, the right eye and the mouth. We compute the center of those components as the average of the corner positions.

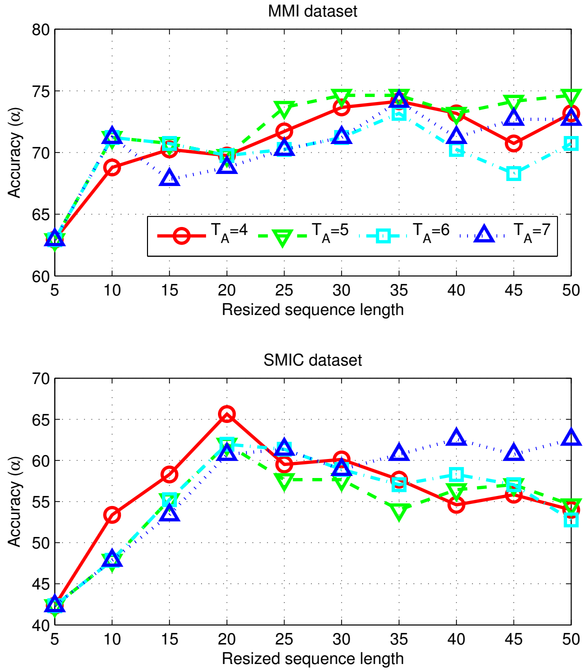


Fig. 12. Performance with respect to (resized) sequence length T indicates sensitivity to frame rate, as the apparent motion speed changes when a sequence is resized temporally. Results are obtained with dynamic bases only.

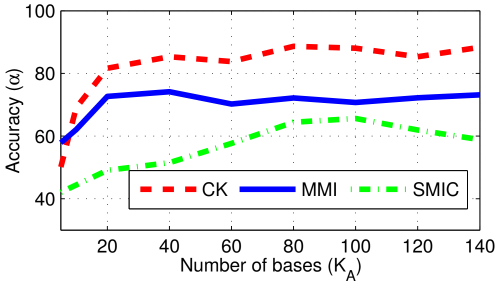


Fig. 13. Performance of the dynamic features our method with respect to the number of bases K_A on the CK+, MMI and SMIC datasets.

TABLE IV
PERFORMANCE OF OUR METHOD ON CK+, MMI AND SMIC, WHEN BASES ARE LEARNT IN A WITHIN-DATABASE MANNER.

Test Dataset	K_A, K_B	Performance with Static Features	Performance with Dynamic Features	Performance with Both Features Types
CK+	100, 100	94.81	89.01	**96.02**
MMI	60, 60	57.56	73.66	**75.12**
SMIC	100, 100	17.79	**65.64**	60.74

We analyse sensitivity to frame rate by experimenting on test sequences that are resized to various numbers of frames T. Whenever unspecified, T is set as $T = 8$ for CK+ and $T = 20$ for MMI and SMIC. We set $H_A = 6$ for the tests on CK+ and MMI, and $H_A = 12$ for SMIC. The parameter T_A is set based on the sequence length as $T_A = \lfloor \frac{T}{5} \rfloor$. T_B is set as $T_B = 1$.

The training for all the facial components (*i.e.* left eye, right eye, and mouth) takes approximately 90 minutes in total (MATLAB implementation running on a laptop with an Intel-$i5$ CPU). The average computation time for our representation is 0.432 seconds per frame. The bottleneck in this process is the computation of the Gabor coefficients (0.354 seconds). Once the Gabor coefficients are obtained, computing the dynamic coefficients, $\mathbf{u}$, takes 0.042 seconds and computing the static coefficients, $\mathbf{v}$, takes 0.036 seconds. For comparison, the average computation time of the standard LBP-TOP [8] representation on the same sequences is 0.023 seconds per frame. The MATLAB® code of our method is provided in ftp://spit.eecs.qmul.ac.uk/pub/es/supp.zip.

D. Discussion

We first analyse how the frame rate of test sequences and the number of bases affect performance. During these tests we use only dynamic bases. Then, we compare the performance of our method with that of state-of-the-art dynamic representations.

The length of the original MMI sequences varies from 32 to 244 frames. Fig. 12 (top) shows how performance varies on the MMI dataset when the sequences are downsampled to various lengths T. We report performance for various temporal pooling windows T_A, as the optimal value of this parameter may depend on the sequence length. The lowest performance occurs when test sequences are resized to 5 frames. There is limited variation when sequences are resized to 20 frames or longer, which suggests that the performance has little sensitivity to the frame rate of the sequences that are used while learning the bases. The best performance is not attained when T takes the value used while learning the bases (*i.e.* $T = 10$, see Section VI-C). The original SMIC sequences vary between 13 and 60 frames. The performance on SMIC becomes particularly low when sequences are downsampled to short lengths such as $T = 5$ frames. The micro-expressions in SMIC are fleeting, and therefore difficult to recognise when the frame rate is too low [20]. However, the performance of our method shows little variation for sequences of $T = 20$ frames or longer. This suggests that the proposed method has little sensitivity to frame rate variations when recognising micro-expressions, given that the frame rate is not too low.

Fig. 13 shows the performance variation with respect to the number of bases, K_A. The performance saturates with relatively small K_A values for the CK+ and MMI datasets, such as $K_A = 40$, and there is little improvement, or even a decrease in performance, for larger K_A values. Higher values such as $K_A = 80$ or $K_A = 100$ achieve better performance on the SMIC dataset. Table IV lists the best results obtained by our method on all datasets for within-database learning with LOSO cross-validation, and reports the performance of static features as well. Static features are sufficient to achieve high performance on the CK+ dataset. This is not surprising as other static representations (*e.g.* [49], [15]) achieve similar performance on this dataset. The dynamic features are useful on the more challenging MMI and SMIC datasets. The high performance achieved with dynamic features on SMIC is consistent with the findings in psychology that highlight the importance of temporal variation for recognising subtle expressions [6].

Finally, we report results on all datasets with a unified

TABLE V
CLASSIFICATION ACCURACY ON CK+, MMI AND SMIC. THE 'WITHIN-DATASET' COLUMN REFERS TO THE CONDITION WHEN THE TEST DATASET IS USED BOTH TO LEARN THE REPRESENTATION AND TO SET ITS PARAMETERS. THE (OPTIONAL) SECOND REFERENCE REFERS TO THE SOURCE THAT THE RESULTS ARE COLLECTED FROM.

Ref.	Method	Engineered	Learnt	Needs Training Labels	Within-dataset Validation	Cross-dataset Validation	Accuracy on **CK+** (α)	Accuracy on **MMI** (α)	Accuracy on **SMIC** (α)
[50]	CFD-WL	✓			N/A	N/A	92.32	–	–
[8],[12]	LBP-TOP	✓			N/A	N/A	88.99	59.51	–
[51],[12]	3D-HOG	✓			N/A	N/A	91.44	60.89	–
[52],[12]	3D-SIFT	✓			N/A	N/A	81.35	64.39	–
[53]	Optical strain	✓			N/A	N/A	–	–	53.56
[54]	STLBP-IP	✓			N/A	N/A	–	–	57.93
[55]	AdaBst+STM	✓			N/A	N/A	–	–	44.34
[8],[20]	LBP-TOP	✓			N/A	N/A	–	–	49.30
[56]	ITBN	✓			N/A	N/A	86.30	59.70	–
[10]	DTAGN		✓	✓	✓		**96.94**	*66.33*	–
[13]	3DCNN-DAP		✓	✓	✓		92.40	63.40	–
[12]	Expressionlets		✓		✓		91.13	65.37	–
[12]	Expressionlets & discr. learning		✓	✓	✓		94.19	**75.12**	–
Proposed: F-Bases			✓		✓		*96.02*	**75.12**	**65.64**
			✓			✓	89.29	–	*60.36*

representation — a representation learnt from a specific dataset for a fixed K_A value. To have a unified representation that is relatively compact and achieves good performance on both large- and small-intensity expressions, we set $K_A = 60$. We train the unified representation on MMI, which is the most comprehensive of the three datasets as it includes the onset, apex and offset phases (see Table III).

We compare with state-of-the-art dynamic representations that were validated on the CK+, MMI and SMIC datasets. We consider only the studies that used the entire sequences on the MMI dataset without using the manually annotated apex frames. The learnt representations that we compare with on the CK+ and MMI datasets are Expressionlets [12], DTAGN [10] and 3DCNN-DAP [13] (see Table I for the extensions of the abbreviations). We further compare with ITBN [56], a method that proposes semantic modelling of expressions, as well as the (engineered) 3D-HOG [51], 3D-SIFT [52] and LBP-TOP [8] representations. We take the results reported in the papers.

Table V reports the results of the methods under analysis on all three datasets. The other learnt representations are validated through within-dataset experiments, i.e. the representations are trained and tested on the same dataset, with different learning parameters for each dataset. We also report results for within-database validation and the cross-database validation results by using the representation learnt on MMI for testing on CK+ and SMIC. DTAGN attains the best accuracy on CK+ and our method achieves comparable results through within-database validation. Most methods achieve high performance (over 90%) on the CK+ dataset, which contains exaggerated expressions with time-aligned sequences (all finish at the apex of the expressions).

Recognition results on MMI are generally lower than those on CK+. Although MMI also contains posed expressions, the fact that the apex frames are not known a priori is a challenge, as an expression is recognised most easily at its apex. Moreover, unlike the CK+ dataset, some of the subjects are wearing glasses, headcloth, or have beard or moustache. Two methods stand out with their high performance on MMI: our method and Expressionlets. However, the latter obtains good results only when the representation is augmented with discriminative learning, which requires a separate training with emotion labels, whereas our method (i) does not require training labels and (ii) can be applied on sequences with labels that are not included in the training set.

On the SMIC dataset, we provide a comparison with LBP-TOP [20], Optical Strain [53], STLBP-IP [54] and a method that uses LBP-TOP with AdaBoost and Selective Transfer Machine (AdaBst+STM) [55]. All these representations are engineered. To the best of our knowledge, there exists no learnt representation tested on the SMIC dataset.

Our method achieves the highest performance on SMIC (rightmost column of Table V), using within-dataset validation (with nearly a 7% improvement compared to other methods) and with cross-database validation, i.e. testing with the representation that was trained and optimised on MMI. This highlights the generalisation ability of our representation: The training dataset (MMI) contains sequences of posed expressions recorded with relatively low temporal resolution (∼25 fps), whereas the test dataset (SMIC) includes sequences of spontaneous expressions recorded with higher resolution (100 fps). Moreover, the MMI dataset includes 6 classes of pronounced expressions, whereas the SMIC dataset contains 3 classes of subtle expressions.

In summary, the proposed method achieves state-of-the-art or comparable performance when, similarly to other representations, is validated through within-database validation. Moreover, the cross-database results highlight the *generalisation* capabilities of the proposed method, as the same representation achieves comparable performance with other methods even when the training dataset differs from the test dataset in terms of frame rate, temporal phases of expressions, the expression labels, and the intensity of expressions (see Table III).

VII. Conclusion

We proposed a novel dynamic representation for facial expression analysis that characterizes facial expression variations with a linear combination of basis functions corresponding to localised movements. When a sequence is decomposed through this linear model, each basis coefficient enables inference on whether a particular movement exists in the sequence, and the magnitude of the coefficient provides information about the intensity of the movement. With this design the learnt representation efficiently recognises facial expressions across a range of intensities and shows little sensitivity to frame rate. Importantly, unlike other learnt representations, the proposed approach achieves state-of-the-art performance without using the expression labels of training sequences when learning the features. To the best of our knowledge, we proposed the first learnt representation that is designed to model expressions across a range of intensities and is validated in recognising both pronounced *and* micro expressions.

To achieve a more compact description and to address the person-specific biases of the bases (see Section V-A), as future work we will include a layer that learns the relationships among bases (*e.g.* [57]) or use a *bilinear* model [58] that recognises different transformations (*e.g.* spatial shift) of the same basis. Moreover, although for computational tractability we assumed independence between phase shifts and magnitudes, a more complex approach could model the two jointly.

References

[1] Y.-l. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97–115, 2001. 1

[2] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39–58, 2009. 1, 3

[3] M. Pantic, "Machine analysis of facial behaviour: Naturalistic and dynamic behaviour," *Philosophical Trans. Royal Society B: Biological Sciences*, vol. 364, no. 1535, pp. 3505–3513, 2009. 1, 3

[4] H. Gunes and B. Schuller, "Categorical and dimensional affect analysis in continuous input: Current trends and future directions," *Image and Vision Computing*, vol. 31, no. 2, pp. 120 – 136, 2013. 1

[5] E. Sariyanidi, H. Gunes, and A. Cavallaro, "Automatic analysis of facial affect: A survey of registration, representation and recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2015. 1, 2, 3, 7

[6] Z. Ambadar, J. W. Schooler, and J. Cohn, "Deciphering the enigmatic face: The importance of facial dynamics in interpreting subtle facial expressions," *Psychological Science*, vol. 16, no. 5, pp. 403–410, 2005. 1, 2, 11

[7] T. Wu, M. Bartlett, and J. Movellan, "Facial expression recognition using Gabor motion energy filters," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, 2010, pp. 42–47. 1, 2, 3

[8] G. Zhao and M. Pietikäinen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007. 1, 2, 3, 11, 12

[9] B. Jiang, M. Valstar, B. Martinez, and M. Pantic, "Dynamic appearance descriptor approach to facial actions temporal modelling," *IEEE Trans. Systems, Man and Cybernetics – Part B*, vol. 44, no. 2, pp. 161–174, 2013. 1

[10] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," in *Proc. IEEE Int'l Conf. Computer Vision*, 2015, pp. 2983–2991. 1, 2, 12

[11] S. Elaiwat, M. Bennamoun, and F. Boussaid, "A spatio-temporal rbm-based model for facial expression recognition," *Pattern Recognition*, vol. 49, no. C, pp. 152–161, 2016. 1, 2

[12] M. Liu, S. Shan, R. Wang, and X. Chen, "Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 1749–1756. 1, 2, 3, 12

[13] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen, "Deeply learning deformable facial action parts model for dynamic expression analysis," in *Proc. Asian Conf. Computer Vision*, 2014, pp. 143–157. 1, 2, 12

[14] S. Porter, L. Ten Brinke, and B. Wallace, "Secrets and lies: Involuntary leakage in deceptive facial expressions as a function of emotional intensity," *J. of Nonverbal Behavior*, vol. 36, no. 1, pp. 23–37, 2012. 1

[15] E. Sariyanidi, H. Gunes, M. Gökmen, and A. Cavallaro, "Local Zernike moment representations for facial affect recognition," in *Proc. British Machine Vision Conf.*, 2013. 1, 11

[16] P. Ekman, W. Friesen, and J. Hager, *The Facial Action Coding System*, 2nd ed., 2002. 1, 4, 5

[17] L. I. Reed, M. A. Sayette, and J. F. Cohn, "Impact of depression on response to comedy: a dynamic facial coding analysis." *J. Abnormal Psychology*, vol. 116, no. 4, p. 804, 2007. 1

[18] P. Lucey, J. Cohn, I. Matthews, S. Lucey, S. Sridharan, J. Howlett, and K. Prkachin, "Automatically detecting pain in video through facial action units," *IEEE Trans. Systems, Man and Cybernetics – Part B*, vol. 41, no. 3, pp. 664–674, 2011. 1, 3

[19] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *Proc. IEEE Int'l Conf. Multimedia and Expo*, 2005, p. 5. 1, 3, 7, 8, 9

[20] X. Li, T. Pfister, X. Huang, G. Zhao, and M. Pietikäinen, "A spontaneous micro facial expression database: Inducement, collection and baseline," in *IEEE Int'l Conf. Face and Gesture Recognition*, 2013, pp. 1–6. 1, 2, 9, 10, 11, 12

[21] R. Zhi, M. Flierl, Q. Ruan, and W. Kleijn, "Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition," *IEEE Trans. Systems, Man and Cybernetics – Part B*, vol. 41, no. 1, pp. 38–52, 2011. 2

[22] P. Khorrami, T. L. Paine, and T. S. Huang, "Do deep neural networks learn facial action units when doing expression recognition?" *arXiv preprint arXiv:1510.02969*, 2015. 2

[23] M. Ranzato, J. Susskind, V. Mnih, and G. Hinton, "On deep generative models with applications to recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 2857–2864. 2

[24] D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *Int'l J. Computer Vision*, vol. 5, no. 1, pp. 77–104, 1990. 2, 4

[25] C. Cadieu and B. Olshausen, "Learning intermediate-level representations of form and motion from natural movies," *Neural computation*, vol. 24, no. 4, pp. 827–866, 2012. 2, 5, 6, 10

[26] S. Rifai, Y. Bengio, A. Courville, P. Vincent, and M. Mirza, "Disentangling factors of variation for facial expression recognition," in *Proc. European Conf. Computer Vision*, 2012, pp. 808–822. 2

[27] S. Cotter, "Sparse representation for accurate classification of corrupted and occluded facial expressions," in *IEEE Int'l Conf. Acoustics Speech and Signal Processing*, 2010, pp. 838–841. 2

[28] P. Lucey, J. 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, 2010, pp. 94 – 101. 3, 7, 9

[29] T. Pfister, X. Li, G. Zhao, and M. Pietikäinen, "Recognising spontaneous facial micro-expressions," in *Proc. IEEE Int'l Conf. Computer Vision*, 2011, pp. 1449–1456. 3

[30] J. F. Cohn, Z. Ambadar, and P. Ekman, "Observer-based measurement of facial expression with the facial action coding system," *The handbook of emotion elicitation and assessment*, pp. 203–221, 2007. 3

[31] S.-J. Vick, B. M. Waller, L. A. Parr, M. C. S. Pasqualini, and K. A. Bard, "A cross-species comparison of facial morphology and movement in humans and chimpanzees using the facial action coding system (FACS)," *J. Nonverbal Behavior*, vol. 31, no. 1, pp. 1–20, 2007. 3

[32] J. A. Coan and J. J. Allen, *Handbook of emotion elicitation and assessment*. Oxford Univ. Press, 2007. 3

[33] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data–recommendations for the use of performance metrics," in *Proc. Int'l Conf. Affective Computing and Intelligent Interaction*, 2013, pp. 245–251. 3

[34] H.-Y. Wu, M. Rubinstein, E. Shih, J. V. Guttag, F. Durand, and W. T. Freeman, "Eulerian video magnification for revealing subtle changes in the world." *ACM Trans. Graph.*, vol. 31, no. 4, p. 65, 2012. 4

[35] L. Xu, J. Chen, and J. Jia, "A segmentation based variational model for accurate optical flow estimation," in *Proc. European Conf. Computer Vision*, 2008, pp. 671–684. 4

[36] B. A. Olshausen, C. F. Cadieu, and D. K. Warland, "Learning real and complex overcomplete representations from the statistics of natural images," in *Optical Engineering Applications*, 2009, pp. 74 460S–74 460S. 4

[37] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996. 4, 10

[38] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997. 4, 5, 6

[39] A. Hyvärinen, J. Hurri, and J. Väyrynen, "Bubbles: a unifying framework for low-level statistical properties of natural image sequences," *J. Optical Society of America A*, vol. 20, no. 7, pp. 1237–1252, 2003. 5

[40] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Machine Learning Research*, vol. 5, pp. 1457–1469, 2004. 5, 7, 10

[41] I. Nabney, *NETLAB: algorithms for pattern recognition*. Springer Science & Business Media, 2002. 6

[42] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988. 6

[43] A. Y. Yang, S. S. Sastry, A. Ganesh, and Y. Ma, "Fast l1-minimization algorithms and an application in robust face recognition: A review," in *Proc. IEEE Int'l Conf. Image Processing*, 2010, pp. 1849–1852. 7

[44] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. 10

[45] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, 2013. 10

[46] E. Sariyandi, H. Gunes, and A. Cavallaro, "Probabilistic temporal subpixel registration for facial expression analysis," in *Proc. Asian Conf. Computer Vision*, 2014. 3, 10

[47] D. Kato and I. Ohzawa. (2012) 2D Gabor wavelet transform and inverse transform demo using matlab. [Online]. Available: https://visiome.neuroinf.jp/modules/xoonips/detail.php?item_id=6951 10

[48] M. Schmidt. (2005) minfunc: unconstrained differentiable multivariate optimization in MATLAB. [Online]. Available: http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html 10

[49] K. Sikka, T. Wu, J. Susskind, and M. Bartlett, "Exploring bag of words architectures in the facial expression domain," in *Proc. European Conf. Computer Vision Workshops*, 2012, pp. 250–259. 11

[50] X. Huang, G. Zhao, W. Zheng, and M. Pietikäinen, "Towards a dynamic expression recognition system under facial occlusion," *Pattern Recognition Letters*, vol. 33, pp. 2181–2191, 2012. 12

[51] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. British Machine Vision Conf.*, 2008, pp. 275–1. 12

[52] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," in *Proc. ACM Conf. Multimedia*, 2007, pp. 357–360. 12

[53] S.-T. Liong, R. C.-W. Phan, J. See, Y.-H. Oh, and K. Wong, "Optical strain based recognition of subtle emotions," in *Proc. IEEE Int'l Conf. Intelligent Signal Processing and Communication Systems*, 2014, pp. 180–184. 12

[54] X. Huang, S.-J. Wang, G. Zhao, and M. Pietikäinen, "Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection," in *Proc. IEEE Int'l Conf. Computer Vision Workshops*, 2015. 12

[55] A. C. Le Ngo, R. C.-W. Phan, and J. See, "Spontaneous subtle expression recognition: Imbalanced databases and solutions," in *Proc. Asian Conf. Computer Vision*, 2014, pp. 33–48. 12

[56] Z. Wang, S. Wang, and Q. Ji, "Capturing complex spatio-temporal relations among facial muscles for facial expression recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 3422–3429. 12

[57] Y. Karklin and M. S. Lewicki, "A hierarchical bayesian model for learning nonlinear statistical regularities in nonstationary natural signals," *Neural computation*, vol. 17, pp. 397–423, 2005. 13

[58] D. B. Grimes and R. P. Rao, "Bilinear sparse coding for invariant vision," *Neural computation*, vol. 17, no. 1, pp. 47–73, 2005. 13