Spatio-temporal Representation and Analysis of Facial Expressions with Varying Intensities

by

Evangelos Sariyanidi

BE in Control Engineering 2009

MSc in Control and Automation Engineering 2012

A dissertation submitted to

The School of Electronic Engineering and Computer Science

in partial fulfilment of the requirements for the Degree of

Doctor of Philosophy

in the subject of

Electronic Engineering

Queen Mary University of London

Mile End Road

E1 4NS, London, UK

November 2017

I, Evangelos Sariyanidi, confirm that the research included within this thesis is my own work, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature: 

Date: 1 November 2017

To my family

Thesis supervisors

Dr. Hatice Gunes

Prof. Andrea Cavallaro

Author

Evangelos Sariyanidi

Abstract

Facial expressions convey a wealth of information about our feelings, personality and mental state. In this thesis we seek efficient ways of representing and analysing facial expressions of varying intensities. Firstly, we analyse state-of-the-art systems by decomposing them into their fundamental components, in an effort to understand what are the useful practices common to successful systems. Secondly, we address the problem of sequence registration, which emerged as an open issue in our analysis. The encoding of the (non-rigid) motions generated by facial expressions is facilitated when the rigid motions caused by irrelevant factors, such as camera movement, are eliminated. We propose a sequence registration framework that is based on pre-trained regressors of Gabor motion energy. Comprehensive experiments show that the proposed method achieves very high registration accuracy even under difficult illumination variations. Finally, we propose an unsupervised representation learning framework for encoding the spatio-temporal evolution of facial expressions. The proposed framework is inspired by the Facial Action Coding System (FACS), which predates computer-based analysis. FACS encodes an expression in terms of localised facial movements and assigns an intensity score for each movement. The framework we propose mimics those two properties of FACS. Specifically, we propose to learn from data a linear transformation that approximates the facial expression variation in a sequence as a weighted sum of localised basis functions, where the weight of each basis function relates to movement intensity. We show that the proposed framework provides a plausible description of facial expressions, and leads to state-of-the-art performance in recognising expressions across intensities; from fully blown expressions to micro-expressions.

Contents

List of Figures

List of Tables

List of Abbreviations

3DCNN-DAP	3D CNN Deformable Action Parts	22
AAM	Active Appearance Models	10
AU	Action Unit	3
AVEC	Audio/Visual Emotion Challenge	8
BoW	Bag-of-Words	22
CLM	Constraint Local Model	13
CRF	Conditional Random Field	27
DBN	Dynamic Bayesian Network	27
DTAGN	Deep Temporal Appearance-Geometry Network	22
FACS	Facial Action Coding System	3
FERA	Facial Expression Recognition and Analysis	8
FFT	Fast Fourier Transform	12
FPR	False Positive Rate	41
GP-NMF	Graph-Preserving NMF	16
GradCorr	Gradient Correlation	12
HMM	Hidden Markov Model	26
HoG	Histogram of Gradients	14
IC	Independent Component	21
ITBN	Interval Temporal Bayesian Network	75
LBP	Local Binary Patterns	14
LBP-TOP	LBP from Three Orthogonal Planes	19
LDA	Linear Discriminant Analysis	25
LK	Lucas-Kanade	12
LOSO	Leave One Subject Out	72
LPQ	Local Phase Quantisation	14

LPQ-TOP	LPQ from Three Orthogonal Planes	19
MUMIE	Multiple Regressors for Misalignment Estimation	31
NMF	Non-negative Matrix Factorisation	13
PCA	Principal Component Analysis	25
QLZM	Quantised Local Zernike Moments	14
RBM	Restricted Boltzmann Machine	22
R-FFT	Robust Fast Fourier Transform	46
ROC	Receiver Operating Characteristics	41
RVM	Relevance Vector Machine	27
SD-NMF	Subclass Discriminant NMF	16
SVM	Support Vector Machine	26
TPR	True Positive Rate	41

Acknowledgements

First and foremost, my thanks go to my family. I knew that I can always rely on their support in difficult times if needed, which gave me the freedom to search my way and do what I aspire the most — a very privileged position to be in. My family had always the highest hopes for me, and did everything in their power to ensure that I have all the necessary support and tools to fulfil my aspirations. Ευχαριστώ για τα πάντα.

I am all grateful to my supervisors, Hatice Gunes and Andrea Cavallaro, for all their ideas and insights, their enthusiasm in our work and their encouraging me towards more ambitious goals. They have patiently revised countless drafts of our papers, and created time even in their most busy schedule. Thir support helped me to grow in multiple dimensions, beyond research. My thanks also to Gianluca Monaci for three very exciting and productive months of internship in Philips Eindhoven, in a very friendly environment. I would also like to thank my former supervisors in Istanbul Technical University; Hakan Temeltaş, who provided me with a great environment to start my research, and Muhittin Gökmen, who let me into his team where we had a most productive experience, crystallising my interest in computer vision.

My warmest thanks to all my friends in the MMV lab and the vision group. Additional thanks are in order for my lab mates who never refused to review drafts of our papers, giving lots of useful insights and advices. Like any fulfilling experience, my four years in London were not without difficult times, but I always knew that “sunshine will follow the rain” — thanks to each and everyone how made that possible.

Chapter 1

Introduction

1.1 Scope of the thesis

Facial expressions are an integral part of our lives. They are a window into our emotions, mental state, mood and personality. Expressions also regulate our day-to-day social communications. Not too surprisingly, the production, perception, interpretation and synthesis of facial expressions have been widely studied in various artistic and scientific disciplines.

The computer-based analysis of facial expressions can enable novel technologies and applications in various domains including healthcare (*e.g.* pain analysis), driving (*e.g.* drowsiness detection), lip reading, animation (*e.g.* facial action synthesis) and social robotics [73, 248]. The automated analysis of expressions can also have a significant impact in cognitive sciences, as the relation between expressions and higher-level personality traits, mental states, or cognitive states are all actively studied problems, and computers capable of quantifying expressions can enable reproducible research on large archives of facial data.

One of the most important questions is, how should we represent the facial expression in a video? To put in another way, how do we convert the image sequence into a numerical representation that maps similar expressions together and different ones far apart? This is a fundamental question for many vision-based applications, and indeed, solutions found for other applications have been influential. One of the most-popular approaches to facial expression representation is using generic local texture or edge descriptors, inspired from their success in other applications, including facial identity recognition [5] or person detection [45]. Recent approaches question the

optimality or practicality of such so-called engineered representations, and use machine learning to devise representations automatically from data. This is hardly a trend unique to facial expression analysis; it is observed from object recognition to text analysis, to pedestrian detection [110].

Solutions inspired directly from other applications will have their limitations, unless they take into account the nature of facial expressions and the way emotions are modelled. An important characteristic of expressions is that they do not occur suddenly; the *intensity* of an expression changes gradually until the expression reaches its apex. Also, the intensity at the apex is not always the same; we can express our happiness with a fully blown smile or with a subtle lip corner movement. Existing systems tend to approach the recognition of pronounced versus subtle expressions as two different problems; however, in our daily life we display expressions at various intensities, and a unified facial representation that covers all of them is plausible. Another characteristic of automatic affect analysis is that the same data can be labelled in different ways. A video of a smiling face can be labelled with 'happiness' if the six-basic emotion model is used, with 'positive' if another discrete emotion model is used [114], or with a vector of real numbers if a continuous emotion model is used [73]. Therefore, a caveat is in order for supervised learning: A representation learnt using the labels of a specific emotion model can be of little use when recognising emotions labelled with another model.

In this thesis we aim to discover efficient ways of representing and analysing facial expressions of varying intensities. We start by conducting a comprehensive literature review where we decompose state-of-the-art affect analysers into their fundamental components. Thus, we aim to gain a better understanding about which component imposes limitations on the system and which are the practices that contribute to its success. Next, we address one of the open issues that emerged in our analysis, namely, the problem of rigid face registration in sequences. Facial expressions generate non-rigid motions, and their analysis is facilitated when the rigid motions that stem from camera, head or body movement are eliminated. Accurate registration is particularly critical for analysing subtle facial movements, which are difficult to identify as they generate little deformation in facial appearance. We propose a novel registration approach that encodes motion (locally) with Gabor motion energy, and then converts motion energy into rigid misalignment parameters with a set of pre-trained regressors. With comprehensive experiments we show that the proposed method is robust to illumination variations, produces little drift error compared to classical registration approaches and achieves very high registration accuracy, which is essential

for learning and identifying subtle (*i.e.* low-intensity) facial movements.

Finally, we present an unsupervised representation learning framework that is capable of encoding facial expressions across intensities. The proposed framework is inspired from the Facial Action Coding System (FACS) [57], which predates computer-based analysis. FACS was developed by psychologists to measure any expression by breaking it down into its constituent localised movements, namely, its Action Units (AUs). Each AU is associated with a score that defines its intensity. These properties enable a compact description, as different facial expressions often contain common localised movements, and intensity scores enable the usage of the same AU to represent a subtle or an pronounced version of the same facial movement. Our unsupervised framework mimics those two properties. Specifically, we propose to learn from data a linear transformation that approximates the facial expression variation in a sequence as a weighted sum of localised basis functions, where the weight of each basis function relates to movement intensity. Since the framework is unsupervised, the representation learnt on a specific set of expression labels (*e.g.* pronounced six basic expressions [167]) can be used on a test set with other expression labels (*e.g.* three classes of micro-expressions [114]). The proposed model is generative, which enables us to synthesise facial expression sequences and discuss the properties of the learnt bases. Experiments show that the proposed method achieves state-of-the-art performance in recognising pronounced expressions *and* micro-expressions. The key idea of representing via localised movements and discerning between their intensity is implemented using local Gabor filters. This idea can be also incorporated into today's popular hierarchical representations, which often contain similar localised differential filters.

1.2 Face perception models in the human vision system

How do we perceive expressions or interpret their meaning? While these are still open research questions [3], there exists a large body of research that can shine light to them. Research suggests that the human vision system has dedicated mechanisms to perceive facial expressions [27, 217], and focuses on three types of facial perception: holistic, componential and configural perception. *Holistic* perception models the face as a single entity where parts cannot be isolated. *Componential* perception assumes that certain facial features are processed individually in the human vision system. *Configural* perception models the spatial relations among facial components (*e.g.* left eye-right eye, mouth-nose). All these perception models might be used when we perceive

expressions [4, 40, 150, 151], and they are often considered complementary [24, 254, 280].

While those three models address spatial perception, the human vision system makes also use of the *temporal variation* of facial appearance throughout an expression [7]. Temporal variation is known to be a fundamental cue for challenging tasks such as identifying subtle expressions [6] and distinguishing between genuine and posed behaviour [234].

1.3 Models of emotions or expressions

Affect recognition systems aim either at recognising the appearance of facial actions or the emotions conveyed by the actions. The former set of systems usually rely on the FACS [57]. FACS consists of facial AUs, which are codes that describe certain facial configurations (*e.g.* AU 12 is lip corner puller). The production of a facial action has a temporal evolution, which plays an important role in interpreting emotional displays [6, 7]. The temporal evolution of an expression is typically modelled with four temporal segments [57]: neutral, onset, apex and offset. *Neutral* is the expressionless phase with no signs of muscular activity. *Onset* denotes the period during which muscular contraction begins and increases in intensity. *Apex* is a plateau where the intensity usually reaches a stable level; whereas *offset* is the phase of muscular action relaxation. Although the order of these phases is usually neutral-onset-apex-offset, alternative combinations such as multiple-apex actions are also possible [36]. AUs and temporal segments are well-analysed in psychology and their recognition enables the analysis of sophisticated emotional states such as pain [129] and helps distinguishing between genuine and posed behaviour [234].

The systems that recognise emotions consider basic or non-basic emotions. *Basic emotions* refer to the affect model developed by Ekman and his colleagues, who argued that the production and interpretation of certain expressions are hard-wired in our brain and are recognised universally (*e.g.* [56]). The emotions conveyed by these expressions are modelled with six classes: Happiness, sadness, surprise, fear, anger and disgust. Basic emotions are believed to be limited in their ability to represent the broad range of everyday emotions [73]. More recently researchers considered *non-basic emotion* recognition using a variety of alternatives for modelling non-basic emotions. One approach is to define a limited set of emotion classes (*e.g.* relief, contempt) [11]. Another approach, which represents a wider range of emotions, is continuous modelling using affect dimensions [73]. The most established affect dimensions are arousal, valence, power and expectation [73].

1.4 Contributions

The main contributions of this thesis are as follows.

1. We present a comprehensive literature review by breaking down state-of-the-art affect analysers into their fundamental components, namely registration, representation, dimensionality reduction and recognition. Our in-depth analysis exposes open issues and useful practices with the aim of facilitating the design of real-world affect recognition systems.

2. We propose a novel sequence registration framework; we show that, in iterative registration, misalignment can be estimated effectively with pre-trained regressors of Gabor motion energy and that these regressors can generalise and perform accurately on data with illumination variations even when trained with controlled data.

3. We develop the closed-form mathematical expressions that can be used to study the motion perception model of Adelson and Bergen [2] for 2D motion. Specifically, we provide the formulation of *Gabor motion energy* for a *moving line* and show how to tune a spatio-temporal Gabor filter pair to a specific type of motion.

4. We propose an analytically validated normalisation scheme that reduces the sensitivity of Gabor motion energy to temporal illumination variations.

5. We show that the L_2 norm of Gabor motion energy can be used to train multiple regressors with different granularities and also to efficiently perform coarse-to-fine registration with these regressors.

6. We propose a novel facial representation learning framework that is designed for analysing expressions at a range of intensities, and has been validated for recognising both pronounced expressions *and* micro-expressions.

7. 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.

1.5 List of publications

Journal Papers

[J1] E. Sariyanidi, H. Gunes, A. Cavallaro. “Learning Bases of Activity for Facial Expression Recognition”, *IEEE Trans. on Image Processing*, vol. 26, no. 4, pages 1965–1978, 2017

[J2] E. Sariyanidi, H. Gunes, A. Cavallaro. “Robust Registration of Dynamic Facial Sequences”, *IEEE Trans. on Image Processing*, vol. 26, no. 4, pages 1708-1722, 2017

[J3] E. Sariyanidi, H. Gunes, A. Cavallaro. “Biologically-Inspired Motion Encoding for Robust Global Motion Estimation”, *IEEE Trans. on Image Processing*, vol. 26, no. 3, pages 1521–1535, 2017

[J4] E. Sariyanidi, H. Gunes, A. Cavallaro. “Automatic analysis of facial affect: A survey of registration, representation, and recognition”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pages 1113-1133, 2015

Book Chapter

[B1] E. Sariyanidi, H. Gunes, A. Cavallaro. "The role of registration and representation in facial affect analysis", *Handbook of Affective Sci. in Human Factors and HCI* (In press)

Conference Papers

[C1] H. Gunes, O. Celiktutan, E. Sariyanidi, E. Skordos. “Real-time prediction of user personality for social human-robot interactions: Lessons learned from public demonstrations”, *Proc. IEEE Int’l Conf. on Intelligent Robots and Systems Workshops*, 2015

[C2] O. Celiktutan, E. Sariyanidi, H. Gunes. “Let me tell you about your personality!: Real-time personality prediction from nonverbal behavioural cues”, *Proc. IEEE Int’l Conf. on Automatic Face and Gesture Recognition Workshops*, 2015

[C3] O. Celiktutan, Florian Eyben, E. Sariyanidi, H. Gunes, Björn Schuller. “MAPTRAITS 2014: The first audio/visual mapping personality traits challenge”, *Proc. ACM Int’l Conf. on Multimodal Interaction Workshops*, 2014, pp. 529-530

[C4] E. Sariyanidi, H. Gunes, A. Cavallaro. “Probabilistic Subpixel Temporal Registration for Facial Expression Analysis”, *Proc. Asian Conf. on Computer Vision*, 2014, pp. 320-335

[C5] E. Sariyanidi, H. Gunes, M. Gökmen, A. Cavallaro. “Local Zernike Moment Representation for Facial Affect Recognition”, *Proc. British Machine Vision Conf.*, 2013

1.6 Organisation of the thesis

This thesis is organised as follows:

Chapter 1 presents the scope and the contributions of this thesis, and introduces the background regarding the facial perception mechanisms in the human vision system as well as the models of emotion used commonly in psychology.

Chapter 2 presents the literature review of facial affect analysis systems by decomposing them into their fundamental components, namely registration, representation, dimensionality reduction and recognition.

Chapter 3 presents the proposed rigid registration framework. We first discuss the motion energy that is encoded with speed- and orientation-selective Gabor filters, and then describe how to convert motion energy into misalignment parameters and how to handle registration failures.

Chapter 4 presents the proposed unsupervised facial representation learning framework and the optimisation process for implementing the framework. We also discuss the movements encoded in the learnt bases by producing synthetic sequences and also analysing real sequences, and describe how to use the bases for automatic expression recognition.

Chapter 5 concludes the thesis with a summary of achievements, limitations and proposed future directions.

Chapter 2

State of the art

2.1 Introduction

In this chapter we provide a comprehensive analysis of the state of the art by decomposing existing system into their fundamental processes, namely, registration, representation, dimensionality reduction and recognition (see Fig. 2.1). Through this decomposition we aim to gain a better understanding about which process imposes limitations to a system or improves its performance. Each of those processes can be implemented with techniques from several categories, as we depict in Fig. 2.1.

Arguably, the process where most research focused on is facial representation, which can be categorised as spatial or spatio-temporal. Spatial representations encode image sequences frame-by-frame, whereas spatio-temporal representations consider a neighbourhood of frames. Another classification is based on the type of information encoded in space: appearance or shape. Appearance representations use textural information by considering the intensity values of the pixels, whereas shape representations ignore texture and describe shape explicitly.

The recent developments in deep learning [110] challenge the traditional pipeline illustrated in Fig. 2.1, as in deep learning the components of registration, representation dimensionality reduction and recognition can all be performed implicitly in the internal layers of the architecture, or some of the components can be rendered redundant. An example to the latter is the role of registration: some deep learning models work with only roughly registered images; those models can in fact be deliberately trained with not very well-registered images to prevent over-

fitting (*e.g.* [175]). The chief property of deep learning pipelines is to *learn* a representation and to integrate the representation with the classifier/regressor. In the process of learning representation deep models also often perform dimensionality reduction, for example, through pooling [22] or autoencoders [246]. We will discuss deep learning methods separately for spatial (see Section 2.4.6) and for spatio-temporal representations (see Section 2.5.6). We will place a particular emphasis on the latter, as this thesis proposes also a learnt spatio-temporal representation (Chapter 4). We will highlight the useful practices and limitations in the state of the art in Section 2.8, where we will also briefly discuss where deep learning stands in the state of the art.

The main challenges in automatic affect recognition are head-pose variations, illumination variations, registration errors, occlusions and identity bias. Spontaneous affective behaviour often involves *head-pose variations*, which need to be modelled before measuring facial expressions. *Illumination variations* can be problematic even under constant illumination due to head movements. Registration techniques usually yield *registration errors*, which must be dealt with to ensure the relevance of the representation features. *Occlusions* may occur due to head or camera movement, or accessories such as scarves or sunglasses. Dealing with *identity bias* requires the ability to tell identity-related texture and shape cues apart from expression-related cues for subject-independent affect recognition. While being resilient to these challenges, the features of a representation shall also enable the detection of *subtle expressions*.

While discussing existing systems, we will discuss how they deal with the above-mentioned challenges, highlight how they relate to the facial perception models introduced in Section 1.2. We further discuss new classifiers and statistical models that exploit affect-specific dynamics by modelling the temporal variation of emotions or expressions, the statistical dependencies among different facial actions and the influence of person-specific cues in facial appearance

2.2 Validation of affect recognition systems

The validation of an affect recognition system depends on which model of emotion or expression is being used (see Section 1.3); the labels of the training/test data and the evaluation metrics are based on this model.

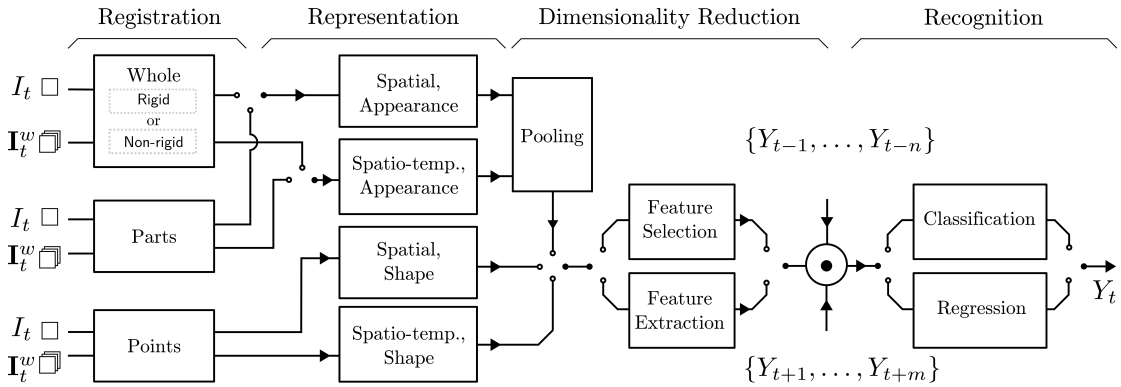


Figure 2.1: The proposed conceptual framework to be used for the analysis and comparison of facial affect recognition systems. The input is a single image (I_t) for spatial representations or a set of frames ($\mathbf{I}_t^w$) within a temporal window w for spatio-temporal representations. The system output Y_t is discrete if it is obtained through classification or continuous if obtained through regression. The recognition process can incorporate previous ($\{Y_{t-1}, \ldots, Y_{t-n}\}$) and/or subsequent ($\{Y_{t+1}, \ldots, Y_{t+m}\}$) system output(s).

2.2.1 Datasets

Most affect recognisers are validated on posed datasets, which differ from naturalistic datasets in terms of illumination conditions, head-pose variations and nature of expressions (subtle vs. exaggerated [73]).

Table 2.1 shows an overview of the datasets used to evaluate affect recognition systems. The CK [96] and MMI [167] datasets are widely used posed datasets and include basic emotion as well as AU annotations. The Enhanced CK dataset [225] provided frame-by-frame AU intensity annotations for the whole CK dataset for 14 AUs and also modified some of the intensity labels that were provided in CK. The CK+ dataset [130] extended CK with spontaneous recordings and novel subjects, annotations and labels (including a non-basic emotion, contempt). A large part of MMI is annotated with temporal segments (neutral, onset, apex, offset). MMI was also extended with new sequences including sequences with spontaneous affective behaviour [236].

There exist non-posed datasets for several affect recognition contexts including categorical basic/non-basic emotion recognition, AU detection, pain detection and dimensional affect recognition. The GEMEP [11] dataset is collected from professional actor portrayals, and includes 12 non-basic emotions and 6 basic emotions. A subset of this database was used in the FERA challenge. Spontaneous AUs can be studied on the public DISFA [141] dataset as well as the partly public M3 (formerly RU-FACS) [16] and UNBC-McMaster [132] datasets. Frame-by-frame AU intensities are provided with DISFA and UNBC-McMaster datasets. Automatic pain recognition can be studied on UNBC-McMaster and COPE datasets [23]. Dimensional affect is studied on

Table 2.1: An overview of the affect recognition datasets.

Dataset	Application and Labels				Statistics and Properties			
	BE	NBE	AU	DA	#Subjects	#Videos	#Images	frame-by-frame labels
CK [96]	6+N	-	✓(+T,+I [225]†)	-	97	486	-	-
GEMEP [11]	6+N	12	✓	-	10	7000	-	-
ISL Frontal-View [225]	-	-	✓+T	-	10	42	-	✓
ISL Multi-View [224]	-	-	✓+T	-	8	40	-	✓
Multi-PIE [71]	3+N	2	-	-	100	-	4200	-
JAFFE [135]	6+N	-	-	-	10	-	213	-
MMI [167, 236]	6+N	-	✓+T	-	75	2420	484	temp.phas.
CK+ [130]	6+N	1	-	-	123	593	-	-
HUMAINE [143]	-	-	-	*A/V**	4	23	-	✓
SEMAINE [143]	3	10††	✓	*A/E/P/V**	150	959	-	✓
RU-FACS [16]	-	-	✓	-	100	100	-	N/A
DISFA [16]	-	-	✓+I	-	27	27	-	✓
Belfast Induced [213]	6+N	Var††	-	*A/V**	256	1400	-	✓
Belfast Naturalistic [53]	4+N	12	-	*A/V**	125	298	-	✓
GENKI-4K [221]	2	-	-	-	N/A	-	4000	N/A
UNBC-McMaster [132]	-	Pain	✓+I	-	25	200	-	✓
COPE [23]	-	Pain	-	-	26	-	204	N/A
SMIC [114]	3†+N	✓	-	-	16	264	-	✓
AFEW [47]	6+N	-	-	-	330	1426		-
SFEW [48]	6+N	-	-	-	N/A			N/A
AM-FED [142]	-	-	12	-	242			✓
FER-2013 [67]	6+N	-	-	-	N/A		35887	N/A
Aff-Wild (images) [275]	-	-	17	-	N/A		10000+	N/A
Aff-Wild (videos) [275]	-	-	-	*A/V**	N/A		500+	✓

†See text for details. ††Refer to the original dataset paper for details.

*These dimensions may be referred to with different names.

BE: Basic emotions; NBE: Non-basic emotions; DA: Dimensional affect;

N: Neutral; +T: Temporal segments; +I: AU intensity;

A: Arousal; *E*: Expectancy; *P*: Power; *V*: Valence

the HUMAINE and SEMAINE datasets.

A problem studied to a lesser extent in affect recognition is the analysis of micro-expressions.

The Spontaneous Micro-expression Corpus (SMIC) [114] can potentially be useful for validating the representations' performance in detecting subtle expressions and replacing the ad-hoc validation procedure used for recognising subtle expressions (*i.e.* recognition at onset, Section 2.5.3). Ground truth is available for 3 emotions, which are clustered from the 6 basic emotions: positive (happiness), negative (anger, fear, disgust and sadness) and surprise.

Recent research efforts also focus on collecting data "in-the-wild" with strong head-pose and illumination variations, and partial occlusions. Popular image-based datasets annotated with the six-basic emotions are the Facial Expression Recognition 2013 (FER-2013) [67] and the Static Faces in the Wild (SFEW) datasets [48]. Acted Facial Expressions In The Wild (AFEW) [47] is a video-based dataset annotated with the six-basic emotions. The Affectiva-MIT Facial Expression Dataset (AM-FED) [142] contains videos annotated with AUs. Aff-Wild [275] contains two datasets: an image dataset annotated with AUs, and a video dataset annotated with continuous arousal and valence labels.

2.2.2 Evaluation

The standard validation protocol is subject independent cross validation. A widely adopted version is leave-one-subject-out (LOSO) cross validation, which enables the researchers to use the maximum data for subject-independent validation. Another validation practice, which highlights the generalisation ability of a method further, is cross-database validation, *i.e.* training is on one dataset and testing on another [92, 101, 225, 237].

Basic emotion recognition has mostly been analysed on posed data, and systems have been evaluated using the average recognition rate or average Area Under the Curve metrics. Although the recognition of posed basic emotions is considered as a solved problem, it is still used for proof of concept of spatial [208, 284] and spatio-temporal representations [127, 258, 266, 267, 281] as well as novel statistical models [33, 186].

AU recognition has been studied both for posed and spontaneous data. The problem is typically formulated as a detection problem and approached by training a 2-class (positive vs. negative) statistical model for each AU. In this setting, results are reported using metrics such as Area Under the Curve, F_1-measure or 2AFC score [91]. A typical problem encountered when evaluating AU performance is imbalanced data, which occurs when the positive AU samples are outnumbered by negative samples, and is particularly problematic for rarely occurring AUs. Jeni *et al.* [90] argue that all above-listed AU metrics are affected negatively by this imbalance. They

suggest to perform skew normalisation to these scores and provide a software to this end [90]. Another AU metric is event agreement [169], which, instead of a frame-by-frame basis, evaluates AUs as temporal events and measures event detection performance. This metric is also extended to Event-F_1 [50] which provides information on not only whether the event is detected or not, but also how successfully the boundaries of the event are identified.

Two well-studied non-basic emotion recognition problems are dimensional affect recognition and pain recognition. In [201], where affect recognition has been performed in terms of quantised affect dimensions, performance has been measured as average recognition rate on four affect dimensions, whereas [200] and [244] considered continuous affect recognition and evaluated performance using the Pearsons' correlation — [244] considered also the recognition of depression and evaluated performance using the mean absolute error and the root mean square error.

In recent years affect recognition competitions emerged as an alternative way to evaluate affect recognition systems. The Facial Expression Recognition and Analysis (FERA)'11 challenge [241] evaluated AU detection and discrete emotion classification for four basic emotions and one non-basic emotion. FERA'15 [239] comprised two sub-challenges: one for AU occurrence identification and another for AU intensity estimation. The Audio/Visual Emotion Challenges (AVEC) [183, 200, 201, 244] evaluated dimensional affect models and also recognition of depression [240, 243]. Affect recognition competitions "in-the-wild" have also been organised, testing the ability of state-of-the-art systems in dealing with difficult head-pose variations, illumination variations or partial occlusions. Two of those challenges are the Kaggle challenge [67] and the EmotiW'15 challenge [49].

2.3 Registration

Face registration is a fundamental step for facial affect recognition. Depending on the output of the registration process, we categorise registration strategies as whole face, part and point registration.

2.3.1 Whole face registration

The region of interest for most systems is the whole face. The techniques used to register the whole face can be categorised as rigid and non-rigid.

Rigid registration

Rigid registration is generally performed by detecting facial landmarks and using their location to compute a global transformation (*e.g.* Euclidean, affine) that maps an input face to a prototypical face. Many systems use the two eye points or the eyes and nose or mouth [91, 120]. The transformation can also be computed from more points (*e.g.* 60-70 points [38]) using techniques such as Active Appearance Models (AAMs) [38]. Computing the transformation from more points has two advantages. First, the transformation becomes less sensitive to the registration errors of individual landmark points. Second, the transformation can better cope with head-pose variations, as the facial geometry is captured more comprehensively.

Non-rigid registration

While rigid approaches register the face as a whole entity, non-rigid approaches enable registration locally and can suppress registration errors due to facial activity. For instance, an expressive face (*e.g.* smiling face) can be warped into a neutral face. Techniques such as AAM are used for non-rigid registration by performing piece-wise affine transformations around each landmark [132]. Alternatively, generic techniques such as SIFT-flow [122] can also be used. The so-called avatar image registration technique [269] adapts SIFT-flow for facial sequence registration. Avatar image registration addresses identity bias explicitly by retaining expression-related texture variations and discarding identity-related variations.

2.3.2 Part-based Registration

A number of appearance representations process faces in terms of parts (*e.g.* eyes, mouth), and may require the spatial consistency of each part to be ensured explicitly. The number, size and location of the parts to be registered may vary (*e.g.* 2 large [225] or 36 small parts [288]).

Similarly to whole face registration, a technique used frequently for parts registration is AAM — the parts are typically localised as fixed-size patches around detected landmarks. Optionally, faces may be warped onto a reference frontal face model through non-rigid registration before patches are cropped (*e.g.* [156, 288]). Alternatively, techniques that perform part detection to localise each patch individually can also be used [279].

2.3.3 Point-based registration

Points registration is needed for shape representations, for which registration involves the localisation of fiducial points. Similarly to whole and parts registration, AAM is used widely for points registration. Alternative facial feature detectors are also used [235, 249]. As localisation accuracy is important for shape representations, it is desirable to validate the feature detectors across facial expression variations [235, 249].

Points in a sequence can be also registered by localising points using a point detector on the first frame and then tracking them. Valstar and Pantic [237] use a Gabor-based point localiser [249] and track the points using particle filter [170].

2.3.4 Sequence registration

So far we discussed how to perform registration spatially. This section discusses the importance of performing (whole-face or part-based) registration for a sequence over time and the methods that can be used for this purpose.

Sequence registration is important for spatio-temporal representations which encode the facial expression variations among subsequent frames. Accurate registration is particularly critical for analysing subtle expressions, which cause little deformation in facial appearance. This issue is illustrated in Fig. 2.2. Two consecutive frames from a facial sequence are shown in Fig. 2.2a. Looking at the frames separately, it is hard to notice any expression, as the human vision system requires to observe the temporal variation in the sequence [6]. A simple way of illustrating the temporal variation is to subtract one image from the other; this difference image is shown in Fig. 2.2b, and highlights the expression that occurs around the eyelids. If, however, one of the images is perturbed even by a translation as small as 0.5 pixels, the difference image now becomes dominated by the registration error and the expression is not seen clearly, as shown in Fig. 2.2d.

A straightforward way to register a sequence is to register each frame independently by first localising facial landmarks within the frame and then performing rigid registration with any of the approaches discussed in Section 2.3.1 or Section 2.3.2. However, jittering errors among consecutive frames is expected in this case, as landmark localisation cannot be performed with very high accuracy (*e.g.* less than one pixel error) in general [29].

An alternative way to sequence registration is to perform (rigid) whole-face or part-based registration only for the first frame, and then to use a generic image pair registration technique

to register the second frame to the first, then the third frame to the second and so on. The rigid registration techniques to use for this purpose can be grouped in three main classes, namely keypoint, transformation-based and direct methods.

Keypoint methods perform registration using sparsely located image points that are centred on visually salient regions with rich texture [260]. While these methods are tolerant to large outlier motions thanks to the use of robust estimators such as Random Sample Consensus (RANSAC) [78], keypoint methods may not perform reliably when outlier motions occur around visually salient regions (*i.e.* regions with texture variations). This occurs with part-based registration or when illumination variations severely reduce the number of matched features [189].

Global transformation-based methods use the invariance properties of the Fourier transform [165, 233], Fourier-Mellin transform [103] or Radon transform [227, 260]. These methods are generally considered to be unsuitable for challenging real-life problems as they are sensitive to outlier motions and illumination variations [230]. Although a robust version of the fast Fourier transform (FFT) [230] is successful against these challenges, its accuracy in simpler conditions without illumination variations can be lower than those of keypoint-based methods [189].

Direct methods minimise an error function of a pair of misaligned frames. The Lucas-Kanade (LK) method minimises the sum of squared difference between two frames and can be rendered partially robust to outliers by dividing frames into blocks [12] or by employing robust estimators [13]. LK methods perform minimisation via gradient descent and may therefore not perform reliably if regions of outlier motions yield high gradient while the remaining regions are relatively flat, which is likely to happen in part-based registration. Extensions of LK differ in the error function that is optimised, the optimisation algorithm or the domain where the optimisation is performed [10, 12, 54, 60, 134, 233]. Methods that operate on the pixel domain are particularly sensitive to illumination variations [233]. Pre-processing with Gabor filters [10] helps improve robustness of LK methods against illumination variations [233]. In a similar manner, performing LK minimisation using other dense features such as LBP or HOG also improves robustness [9]. One of the most robust methods against non-uniform illumination variations is based on the direct maximisation of the gradient correlation coefficient (GradCorr) [233]. GradCorr employs a cosine kernel, which improves robustness against outliers and illumination variations by eliminating local mismatches [233].

Keypoint, transformation-based and direct methods are prone to drift errors in long sequences

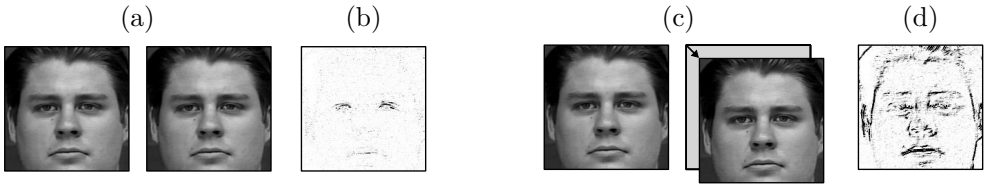


Figure 2.2: An illustration depicting the importance of accurate registration when analysing subtle expressions. (a) Two consecutive images from a sequence that contains a subtle expression around the eyelids, which is hard to notice when looking at static images. These two images are perfectly registered. (b) The temporal variation (*i.e.* difference image) between the perfectly registered images shows correctly the facial activity around the eyelids. (c) The same pair of consecutive images but the second image has been displaced by 0.5 pixels (this displacement is visualised with some magnification to facilitate the interpretation). (d) The temporal variation of the unregistered images is highlighting the registration errors rather than the facial activity even for registration errors as low as 0.5 pixels.

as they register each frame with respect to a reference frame. This problem was highlighted for the LK framework [140] and addressed by a number of methods [140, 164, 199], [8], which were validated on data with limited illumination variations only.

Table 2.2 summarises the methods discussed so far in this section. The method listed in the last row [191] is developed as a part of this thesis' work. We will describe this method in detail throughout Chapter 3, and compare it with a number of techniques in Table 2.2 on sequences with facial expressions, both for whole-face registration *and* for part-based registration.

An alternative to sequence registration can be to use batch alignment methods that are particularly common in registering datasets, such as congealing methods [41, 82]. The advantage of such methods is that they can have low drift error as they register frames jointly, and there exist methods that are robust against illumination variations and gross occlusions thanks to ℓ_1 minimization [171]. However, such methods need to have the entire sequence to register in advance (*i.e.* they are offline).

Finally, although registration and representation are mostly considered as two different components in a facial affect analysis pipeline, it is possible to consider those two as one joint component. The approach of Koelstra *et al.* [101] extracts motion vectors from a sequence in order to apply non-rigid registration, and then uses those vectors for representation (see Section 2.5.4).

2.3.5 Discussion

While some representations (*e.g.* part-based representations) are coupled with a certain type of registration only, others can be used with various registration schemes. For instance, generic appearance representations such as a Gabor representation can be used after performing rigid

Table 2.2: Representative rigid registration methods and how they address illumination variations, drift errors and outlier motions. Key: (K)eypoint, (T)ransformation-based, (D)irect, (S)tatistical Learning.

	Ref.	Approach	Illumination Variations	Drift Errors	Outlier Motions
	[18]	SURF feature matching	Robust features	—	RANSAC
(K)	[161]	MSER feature matching	Robust features	—	RANSAC
	[251]	SIFT feature matching	Robust features	Drift correction	RANSAC
	[165]	Multi-layer Fourier transf.	—	—	—
(T)	[260]	Radon transf.	—	—	—
	[230]	Robust Fourier transf.	Gradient correlation	—	Cosine kernel
	[12]	Lucas-Kanade (LK) matching	—	—	Robust estimator
	[10]	LK matching	Gabor Filtering	—	Robust estimator
(D)	[199]	Robust LK matching	—	Drift correction	Robust estimator
	[164]	Extended LK matching	—	Backgr. modelling	Robust estimator
	[233]	Gradient correlation max.	Gradient correlation	—	Cosine kernel
(S)	[191]	Optimisastion with pre-trained regressors	3D Gabor representation	Multi-frame motion encoding	Pooling, training with noisy data

or non-rigid whole face registration [16, 33] or parts registration [279]. For such representations, the type of information encoded by the overall system depends on the registration strategy employed. More specifically, the registration decides whether configural information will be retained. A non-rigid registration that warps faces to a neutral face may reduce the effect of configural information, or parts registration of individual facial components (*e.g.* eyes, nose and mouth) may neglect configural information completely.

An important decision to be made for registration is how to deal with head-pose variations. While a number of systems approach head-pose as a factor that needs to be suppressed in order to analyse facial activity explicitly [16, 101, 186], others model both facial activity and head-pose simultaneously, arguing that head-pose variations are part of affective behaviour [147, 156, 224]. Indeed, recent studies show that head-pose variation itself is a useful indicator of affective state [1, 75].

Registration is crucial for analysing spontaneous affective interactions, which typically involve head-pose variations. While systems validated on posed data often use simple whole face registration techniques based on 2-4 points, systems validated on spontaneous data rely on more sophisticated whole face, parts or points registration techniques.

AAM is a popular choice to perform whole face, parts or points registration. Although in principle AAM is subject-independent, in practice its accuracy is higher when the model of the subject to register exists *a priori* [70]. A subject-independent alternative is Constrained Local Model (CLM) [188]. However the accuracy of CLMs is generally lower than that of AAMs [33]. The accuracy of both CLM and AAM decreases significantly in naturalistic imaging conditions that include partial occlusions, illumination and head-pose variations [287].

There has been significant progress in developing subject-independent robust landmark localisation techniques in recent years, and a number of techniques that achieve high accuracy in difficult conditions have been proposed [118, 231, 232, 261, 263, 265, 287]. A recently organised facial landmark localisation "in-the-wild" competition has highlighted the progress made [187].

2.4 Spatial representations

Spatial representations encode image sequences frame-by-frame. There exists a variety of *appearance* representations that encode low- or high-level information. Low-level information is typically encoded with low-level histograms and Gabor representations. Higher level informa-

tion is encoded using for example Non-Negative Matrix Factorisation (NMF) or sparse coding. There exist hierarchical representations that consist of cascaded low- and high-level representation layers. Several appearance representations are part-based. *Shape* representations are less common than appearance representations.

2.4.1 Shape representations

The most frequently used shape representation is the facial points representation, which describes a face by simply concatenating the x and y coordinates of a number of fiducial points (*e.g.* 20 [186] or 74 points [133]). When the neutral face image is available, it can be used to reduce identity bias [133] (Fig. 2.3a). This representation reflects registration errors straightforwardly as it is based on either raw or differential coordinate values. Illumination variations are not an issue since the intensity of the pixels is ignored. However, illumination variations may reduce the registration accuracy of the points (Section 2.3.5). Facial points are particularly useful when used to complement appearance representations, as done by the winners of AVEC'12 continuous challenge [156] and FERA'15 AU challenge [204].

Alternative shape representations are less common. One can use the distances between facial landmarks rather than raw coordinates [83]. Another representation computes descriptors specific to facial components such as distances and angles that describe the opening/closing of the eyes and mouth, and groups of points that describe the state of the cheeks [223].

2.4.2 Low-Level histogram representations

Low-level histogram representations (Fig. 2.3b–d) first extract local features and encode them in a transformed image, then cluster the local features into uniform regions and finally pool the features of each region with local histograms. The representations are obtained by concatenating all local histograms.

Low-level features are robust to illumination variations to a degree, as they are extracted from small regions. Also, they are invariant to global illumination variations (*i.e.* gray-scale shifts). Additionally, the histograms can be normalised (*e.g.* unit-norm normalisation [45]) to increase the robustness of the overall representation. These representations are also robust to registration errors as they involve pooling over histograms (Section 2.6.1). Low-level histogram representations are affected negatively by identity bias, as they favour identity-related cues rather than expressions [5, 145, 193]. These representations encode componential information as each

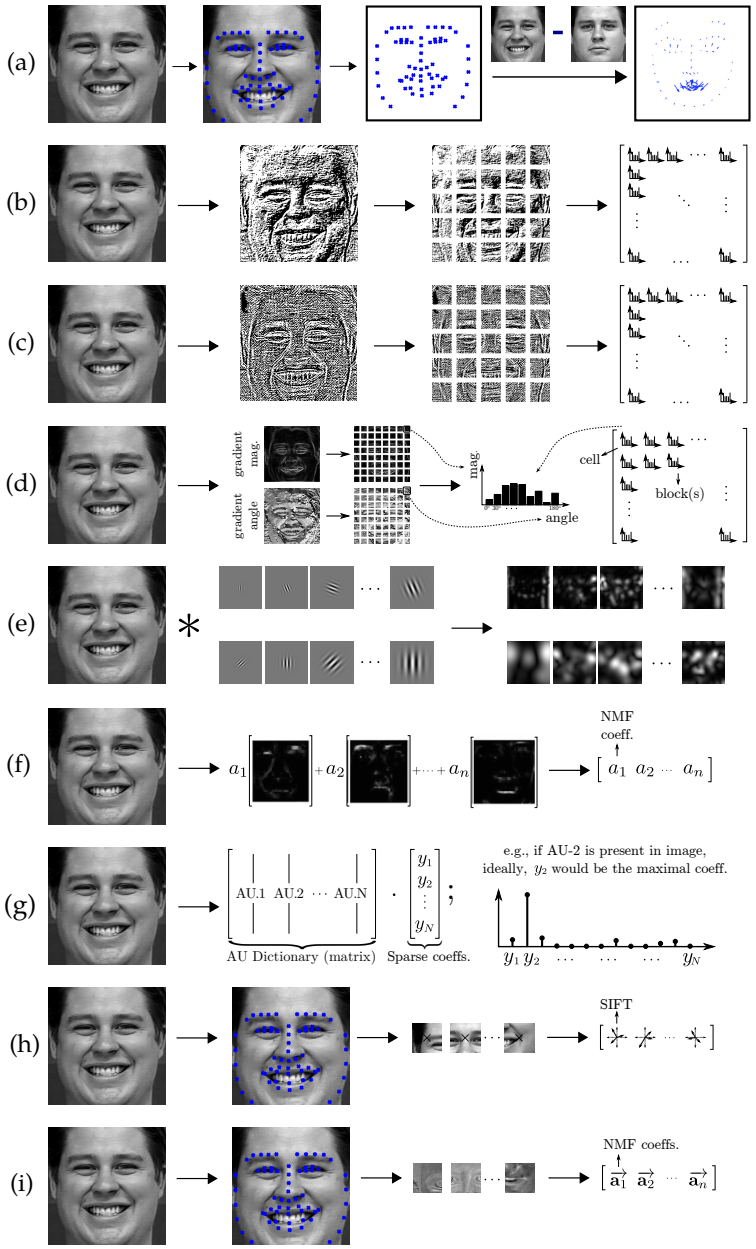


Figure 2.3: Spatial representations. (a) Facial points; (b) LBP histograms; (c) LPQ histograms; (d) HoG; (e) Gabor-based representation; (f) GP-NMF; (g) sparse coding; (h) part-based SIFT; (i) part-based NMF.

histogram describes a region independently from the others. Also, depending on registration (Section 2.3.5), they may implicitly encode configural information, since the global topology of local histograms is retained. Low-level histogram representations are computationally simple and allow for real-time operation [194, 205].

Low level representations, particularly Local Binary Patterns (LBP) [5] and Local Phase Quantisation (LPQ) are very popular. LBP was used by the winner of AVEC'12 word-level challenge [195] and FERA AU detection challenge [204], LPQ was used by prominent systems

in FERA'11 [269] and AVEC'11 [43].

An LBP describes local texture variation along a circular region with an integer [5]. LBP histograms simply count the LBP integers, and therefore the dimensionality of the representation depends on the range of integers. The range of the most common LBP is $[1,256]$. Ahonen *et al.* [5] showed that face images can be represented with a 59-element subset of these patterns (*i.e.* uniform patterns), which operate like edge detectors [259].

The LPQ descriptor was proposed for blur insensitive texture classification through local Fourier transformation [160]. Similarly to an LBP, an LPQ describes a local neighbourhood with an integer ranged in $[1,256]$. Local histograms simply count LPQ patterns, and the dimensionality of each histogram is 256 [160].

The Histogram of Gradients (HoG) approach [45] represents images by the directions of the edges they contain. HoG extracts local features by applying gradient operators across the image and encoding their output in terms of gradient magnitude and angle (Fig. 2.3d). First, local magnitude-angle histograms are extracted from *cells*, and then these local histograms are combined across larger entities (*blocks*) — the dimensionality increases when the blocks are overlapping [45]. HoG was used by a prominent system in the FERA emotion challenge [44].

Another low-level histogram representation is Quantised Local Zernike Moments (QLZM), which describes a neighbourhood by computing its local Zernike moments [194]. Each moment coefficient describes the variation at a unique scale and orientation, and the information conveyed by different moment coefficients does not overlap [220]. The QLZM descriptor is obtained by quantising all moment coefficients into an integer, and the local histograms count QLZM integers.

Low-level representations can be compared from several perspectives. LBP and HoG are compared in terms of sensitivity to registration errors and results suggest that LBP histograms are generally less sensitive [69]. LBP and LPQ are compared in terms of overall affect recognition performance in a number of studies, and LPQ usually outperforms LBP [91, 92, 244, 269]. This may be due to the size of the local description, as LBPs are usually extracted from smaller regions with 3 pixel diameter [205], whereas LPQs are extracted from larger regions of 7×7 pixels [5, 91, 92]. LBPs cause loss of information when extracted from larger regions as they ignore the pixels that remain inside the circular region. On the contrary, LPQ integers describe the regions as a whole. QLZMs also describe local regions as a whole and larger regions such as 7×7 proved more useful, particularly for naturalistic affect recognition [194]. Another comparison

that can be useful for low-level representations is dimensionality. While the local histograms of LBP and LPQ are relatively higher dimensional (due to their pattern size), QLZM and HoG can be tuned to obtain lower-dimensional histograms that proved successful respectively on AVEC data [194] and FERA challenge [44].

2.4.3 Gabor representation

Another representation based on low-level features is the Gabor representation, which is used by various systems including the winner of the FERA'11 AU detection challenge [121, 259] and AVEC'11 [65].

A Gabor representation is obtained by convolving the input image with a set of Gabor filters of various scales and orientations (Fig. 2.3e) [105, 255]. Gabor filters encode componential information, and depending on the registration scheme, the overall representation may implicitly convey configural information (see Section 2.3.5). The high dimensionality of the convolution output renders a dimensionality reduction step essential. As the pixels of Gabor-filtered images contain information related to neighbouring pixels, simple dimensionality reduction techniques such min, max and mean pooling can be used. Gabor filters are differential and localised in space, providing tolerance to illumination variations to a degree [95, 255]. Similarly to low-level histogram representations, Gabor representation suffers from identity bias as it favours identity-related cues rather than expressions [255]. The representation is robust to registration errors to an extent as the filters are smooth and the magnitude of filtered images is robust to small translation and rotations [69, 105]. Robustness to registration errors can be increased further via pooling (Section 2.6.1). Gabor filtering is computationally costly due to convolution with a large number of filters (*e.g.* 40 [255]).

2.4.4 Data-driven representations

All representations discussed so far describe local texture (Fig. 2.3a–e). Implicitly or explicitly, their features encode the distribution of edges. Recent approaches aim instead at obtaining data-driven higher-level representations to encode features that are semantically interpretable from an affect recognition perspective. Two methods that generate such representations are NMF [158, 284] and sparse coding [39, 137, 274]. Alternatively, various feature learning approaches can also be used [184].

NMF methods decompose a matrix into two non-negative matrices. The decomposition is

not unique and it can be designed to have various semantic interpretations. One NMF-based technique is Graph-Preserving NMF (GP-NMF) [284], which decomposes faces into spatially independent components through a spatial sparseness constraint [81]. The decomposition into independent parts encodes componential information, and possibly configural information (see Fig. 2.3f and [284]).

Another NMF-based approach is Subclass Discriminant NMF (SD-NMF) [158], which represents an expression with a multimodal projection (rather than assuming that an expression is unimodally distributed). Unlike GP-NMF, SD-NMF does not explicitly enforce decomposition into spatially independent components. The basis images provided [158] suggest that the information encoded can be holistic, componential or configural.

NMF creates a number of basis images, and the features of NMF-based representations are the coefficients of each basis image (*e.g.* α_1, α_2 in Fig. 2.3f). The method performs minimisation to compute the coefficients, therefore its computational complexity varies based on the optimisation algorithm and the number and size of basis images. Since NMF relies on training, its tolerance against illumination variations and registration errors depends on the training data — the ability of NMF to deal with both issues concurrently is limited as NMF is a linear technique [228]. NMF-based representations can deal with identity bias by learning identity-free basis images (Fig. 2.3f). This depends on the number of identities provided during training as well as the capability of the technique to deal with the inter-personal variation. The dimensionality of NMF-based representations is low — their performance saturates at less than 100 [284] or 200 features [158].

The theory of sparse coding is based on the idea that any image is sparse in some domains, that is, a transformation where most coefficients of the transformed image are zero can be found [28]. The transformation can be adaptive (*e.g.* data-driven) or non-adaptive (*e.g.* Fourier transform), and is based on a so-called dictionary [28]. The flexibility of the dictionary definition gives the researchers the freedom to define dictionaries where the elements of a dictionary are semantically interpretable. In affect recognition, researchers defined dictionaries where each dictionary element corresponds to AUs [137] or basic emotions [39]. The representation is formed by concatenating the coefficients of dictionary elements. In an AU dictionary, the coefficient with the maximal value would ideally point to the AU displayed in the original image (Fig. 2.3g). The coefficients are computed by solving an ℓ_1 minimisation, therefore the computational complex-

ity depends on the optimisation algorithm and the size of dictionary. The representation can be designed to be robust against partial occlusions [39, 274].

An alternative high-level representation paradigm is learning features for multiple tasks concurrently via multi-task learning [184]. One method considered the tasks of face (identity) recognition and facial affect recognition [184] by deriving two independent feature sets — one for each task. The independence assumption can reduce the effect of identity bias, however, it may be a too strong assumption as identity and facial affect cues are often entangled [280].

2.4.5 Part-based representation

Part-based representations process faces in terms of independently registered parts and thereby encode componential information. They discard configural information explicitly as they ignore the spatial relations among the registered parts (Fig. 2.3h,i). Ignoring the spatial relationships reduces the sensitivity to head-pose variation. Part-based representations proved successful in spontaneous affect recognition tasks (*e.g.* AU recognition [89, 288] or dimensional affect recognition) where head-pose variation naturally occurs.

Although most representations can be used in a part-based manner, two representations were explicitly defined so: part-based SIFT [288] and part-based NMF [89].

Part-based SIFT describes facial parts using SIFT descriptors of fixed scale and orientation. The representation inherits the tolerance of SIFT features against illumination variations and registration errors [128]. The dimensionality of the representation is proportional to the number of SIFT descriptors. Part-based SIFT is computationally simple as it only requires the computation of the SIFT descriptors.

Part-based NMF describes facial parts by means of a sparsity-enforced NMF decomposition [89]. An important step in this representation is the removal of person-specific texture details from each patch before the computation of NMF. This step enables the representation to reduce identity bias and place higher emphasis on facial activity (Fig. 2.3i), increasing its potential to deal with subtle expressions. However, texture subtraction may be susceptible to illumination variation and registration errors. Since the representation is based on NMF, its sensitivity against these issues also depends on the training process. The dimensionality of the representation is expected to be low as reducing dimensionality is one of the main motivations behind the use of NMF [89]. The computational complexity mainly depends on the complexity of the NMF algorithm as well as the number of basis matrices and size of each basis matrix. The part-based

NMF representation has been evaluated in terms of the recognition of subtle expressions and shown to outperform spatio-temporal representations [89].

2.4.6 Deep learning

Hierarchical representations become popular in computer vision due to their ability to concurrently address multiple challenges. The leading paradigm for constructing hierarchical representations is deep learning [110]. Thanks to the impressive results it achieved in a variety of difficult pattern recognition problems, such as character, object, face and speech recognition [110], deep learning is now enjoying an increasing popularity in facial affect analysis too [275].

Overall, a representation based on deep learning often contains at least two low-level layers; the first layer convolves the input image with a number of local filters learnt from the data, and the second layer aggregates the convolution output through operations such as pooling [179, 182] (Section 2.6.1). However, deep learning architectures contain increasingly more layers. For example, the widely popular AlexNet [102] contains 8 layers, and another popular architecture includes 27 layers [216].

Deep learning architectures contain a large number of parameters to optimise, and they need large amounts of training data and computation power for their training. However, once trained successfully, they can achieve state-of-the-art performance in computer vision problems in uncontrolled conditions, such as object recognition and face recognition with large viewpoint and lighting variability [110]. Therefore, deep learning architectures are promising candidates for solving the head-pose and illumination variations that typify facial affect recognition in the wild.

Indeed, the successful participants of the recent competitions for affect recognition in the wild typically used deep learning methods. The winner of the Kaggle competition [67] used a deep architecture that replaced the softmax layer with linear SVM [218]. It is remarkable that winner [99], runner up [271] and third system [152] of the static sub-challenge in EmotiW'15 have all used deep learning architectures. It must be noted that all of those challenges are image-based and not video-based. We will return to the latter while discussing spatio-temporal learnt representations (Section 2.5.6) and in our summary (Section 2.8).

To provide the large amounts of required data, deep architectures for facial expression usually use large datasets such as FER-2013 or even combine multiple datasets during training [175]. The optimal depth for facial expression analysis architecture seems to be an open question; while there exist studies that opt for deeper representations [146], a recent study shows that relatively

shallow architectures (*e.g.* 4, 5, 6 layers) may outperform deeper ones (*e.g.* 8, 11 layers) [175].

2.4.7 Discussion

The most notable recent trend is moving from shape to appearance representations and it is mainly due to the low-level representations. The robustness of these representations against generic image processing issues such as illumination variation and registration errors as well as their implementation simplicity had a significant contribution to their popularity. Yet, identity bias remains as an outstanding issue for low-level representations. Identity bias can be reduced in subsequent system layers such as dimensionality reduction (Section 2.6.2) or recognition (Section 2.7.2).

Most representations are sensitive to head-pose variations, therefore may fail in generalising to spontaneous affective behaviour. Although part-based representations reduce the effect of head-pose variations by discarding the spatial relationships among the parts, the appearance of each patch is still affected by the head-pose. Deep learning architectures trained with large amounts of data are also useful for addressing head-pose variations (Section 2.4.6).

Shape representations are crucial for interpreting facial actions [139], and they are not exploited to their full potential. The current state of the art focuses on a small subset of possible shape representations. Firstly, recently used representations are point-based. If we adopt the definition of shape representations as the representations that ignore the intensity value of the pixels, we can see that description through discrete points is not the only option, as one may develop a continuous shape representation (*e.g.* [112, 277]). Secondly, existing representations are vulnerable to registration errors. The state of the art overlooks the possibilities of extracting features that are robust to registration inconsistencies (*e.g.* [66, 277]). Although a small number of systems rely on subspace analysis which may remedy this issue (*e.g.* [136, 186]), most systems rely on absolute or differential point coordinates, which reflect registration errors directly.

A practice that proved particularly useful is using shape representations in conjunction with appearance representations, combining various types of configural, holistic and componential information. This is in accordance with the behaviour of the human vision system when dealing with particularly ambiguous facial displays [24,280] or interpreting different types of expressions [4]. Examples are the system that won the FERA'11 AU sub-challenge, which combined LBP histograms of Gabor images with facial points [203], and the system that won the AVEC'12 fully continuous sub-challenge, which combined componential as well as holistic PCA features with

facial points [156].

2.5 Spatio-Temporal Representations

Spatio-temporal representations consider a range of frames within a temporal window as a single entity, and enable the modelling of temporal variation in order to represent subtle expressions more efficiently. They can discriminate the expressions that look similar in space (*e.g.* closing eyes versus eye blinking [94, 101]), and facilitate the incorporation of domain knowledge from psychology. This domain knowledge relates the muscular activity with higher level tasks, such as distinguishing between posed and spontaneous affective behaviour or recognition of temporal phases (*e.g.* [234, 238]). Most representations are *appearance* representations. The only *shape* representation discussed in this chapter is Geometric Features from Tracked Facial Points.

2.5.1 Geometric features from tracked facial points

This representation aims to incorporate the knowledge from cognitive science to analyse temporal variation and the corresponding muscular activity. It has been used for the recognition of AUs with their temporal phases [237], and the discrimination of spontaneous versus posed smiles [234] and brow actions [238].

The representation describes the facial shape and activity by means of fiducial points [237]. To this end, it uses the raw location of each point, the length and angle of the lines obtained by connecting all points pairwise in space, and the differences obtained by comparing these features with respect to their value in a neutral face. Some of these features describe componential information such as the opening of the mouth, as well as configural information such as the distance between the corner of the eye and the nose (Fig. 2.4a). Other features aim at capturing temporal variation. The temporal window is adjusted according to the video frame rate and the findings of cognitive sciences about neuromuscular facial activity [237]. The representation is computationally simple as it relies on simple operations (*e.g.* subtraction, angle computation).

The representation is sensitive to registration errors as its features are mostly extracted from raw or differential point coordinates. Although the representation describes temporal variation, it may not capture subtle expressions as it is extracted from a small number of facial points (*e.g.* 20 [242]) and depends on accurate point registration. The representation deals with identity bias by including features that describe the deviation from the neutral face. Although the dimensionality

of this representation is relatively small, it risks overfitting as the features are extracted from a much lower number of points [237], therefore, an additional dimensionality reduction scheme is usually applied [237].

2.5.2 Low-level features from orthogonal planes

Extracting features from Three Orthogonal Planes (TOP) is a popular approach towards extending low-level spatial appearance representations to the spatio-temporal domain (Fig. 2.4b,c). This paradigm originally emerged when extending LBP to LBP-TOP [281]. LBP-TOP is applied for basic emotion recognition [281, 282] and AU recognition [91, 92]. Following this method, LPQ is extended to LPQ-TOP ((Local Phase Quantisation from Three Orthogonal Patterns)) and used for AU and temporal segment recognition [91, 92].

As illustrated in Fig. 2.4b, the TOP paradigm extracts features from local spatio-temporal neighbourhoods over the following three planes: The spatial plane (x-y) similarly to the regular LBP, the vertical spatio-temporal plane (y-t) and the horizontal spatio-temporal plane (x-t). Similarly to its spatial counterpart (Section 2.4.2), this representation paradigm extracts local histograms over (spatio-temporal) regions. Therefore, it encodes componential information and, depending on the type of registration, it may implicitly provide configural information. In addition to these, the TOP paradigm encodes temporal variation. For AU recognition, Jiang *et al.* [91] showed that the suitable temporal window can be different for each AU. LBP-TOP and LPQ-TOP are computationally more complex than their static counterparts, however, depending on the size of the spatial and temporal windows of the LBP- or LPQ-TOP operators, real-time processing speed can be achieved [92].

LBP-TOP and LPQ-TOP inherit their robustness against illumination variations from their static counterparts, however, they are more sensitive to registration errors. They assume that texture variations are caused only by facial motion, and therefore they may interpret temporal registration errors as facial activity. The dimensionality of these representations is higher than their static counterparts. While LBP-TOP usually reduces dimensionality by considering only the uniform patterns (*e.g.* 177 patterns per histogram [281]), LPQ-TOP lacks such a concept and the size of possible patterns is larger (*i.e.* 768 per histogram [91, 92]). Both representations are expected to be sensitive to identity bias.

Experiments show that LBP-TOP and LPQ-TOP outperform their spatial counterparts, and LPQ-TOP outperforms LBP-TOP in the task of AU recognition [91].

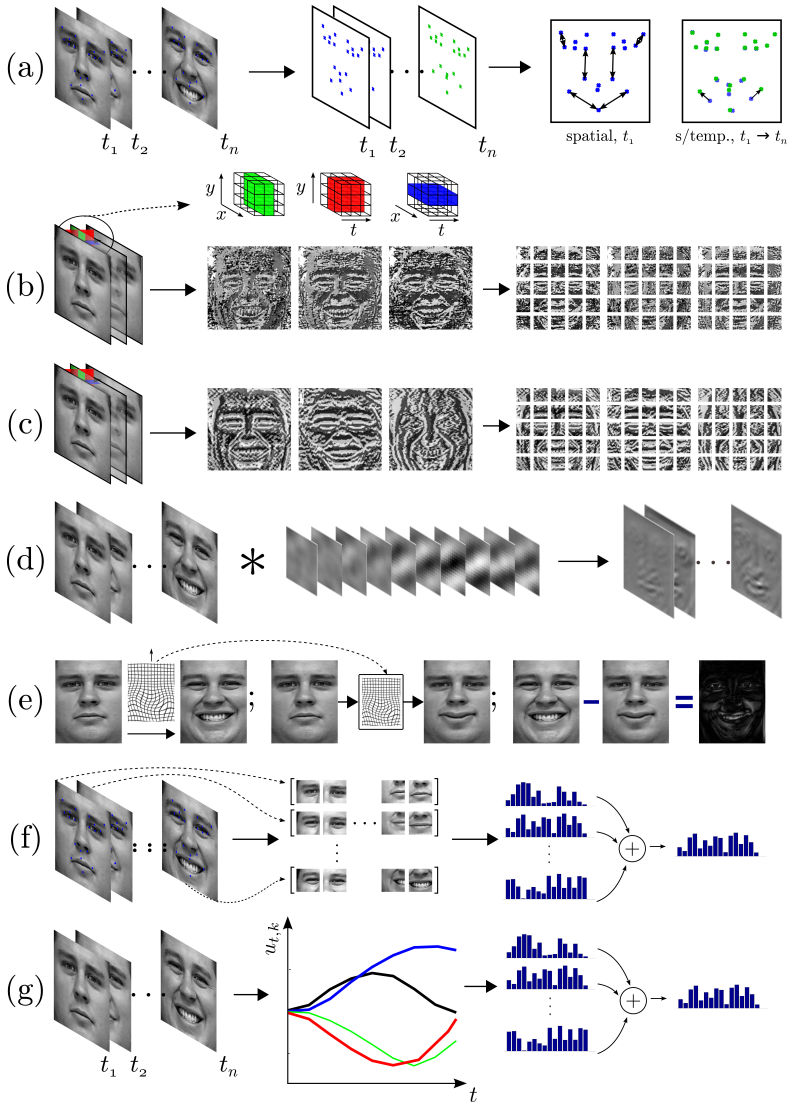


Figure 2.4: Spatio-temporal representations. (a) Geometric features from tracked feature points; (b) LBP-TOP, and the TOP paradigm; (c) LPQ-TOP; (d) spatio-temporal IC filtering, the output on an exemplar spatio-temporal filter; (e) free-form deformation representation, illustration of free-form deformation; (f) temporal BoW; (g) Facial Bases (Chapter 4).

2.5.3 Convolution with smooth filters

An alternative approach for representing the temporal variation in texture with low-level features is applying convolution with smooth spatio-temporal filters (see Fig. 2.4d). Two such approaches are spatio-temporal Gabor filtering [258] and spatio-temporal Independent Component (IC) filtering [127]. Both approaches target explicitly the recognition of subtle expressions.

Gabor and IC filters are localised in space and time. At the spatial level, the output of the filtering encodes componential information. Depending on the registration strategy, the overall representation may also implicitly provide configural information (Section 2.3.5). The main difference between the Gabor and IC filters is that the parameters of Gabor filters are adjusted

manually [258], whereas IC filters are obtained automatically in the process of unsupervised Independent Component Analysis [127]. Both approaches include filters of various temporal windows. The sensitivity of these approaches against illumination variations is expected to be similar to the spatial Gabor filters. However, spatio-temporal Gabor and IC filters are more sensitive to registration errors as they assume temporal registration consistency among successive images in a sequence. The computational overhead of both representations is very high as they involve three-dimensional convolution with a large number of filters (*e.g.* 240 filters [127, 258]). Although the dimensionality of the convolution output is high, straightforward pooling strategies such as min, max and mean pooling [127, 258] can be used.

Gabor and IC representations are used for basic emotion recognition, however, they adopted an unusual validation scheme. Unlike most studies that recognise expressions at the apex phase, these representations aimed at recognising the expressions at early stages (at onset). Spatio-temporal Gabor filters outperform their spatial counterpart [258], and IC filters outperform the manually designed spatio-temporal Gabor filters [127].

2.5.4 Free-Form deformation representation

The free-form deformation representation [101] extends free-form deformation, which is essentially a registration technique, into a representation that extracts features in the process of registration by computing the pixels' spatial and temporal displacement (Fig. 2.4e). This representation is used for AU recognition with temporal segments [101].

Unlike approaches that extract features from uniform subregions, this representation partitions the volumes into non-uniform subregions through quadtree decomposition [101]. This partitioning emphasises regions of high facial activity by allocating to them a larger number of smaller regions. The representation is obtained by extracting a set of spatial and spatio-temporal features (*e.g.* orientation histogram, curl, divergence). These features are extracted independently for each subregion, therefore they can be considered as a form of pooling (Section 2.6.1) that renders the representation robust against small registration errors. The features encode componential information as well as temporal variation.

The spatio-temporal representations discussed so far require temporal registration consistency and rely on external registration techniques to satisfy this. The free-form deformation representation satisfies temporal consistency with its own intrinsic registration layer — free form deformation. Yet, free-form deformation assumes that the head-pose variations of the subject are

limited throughout an image sequence [101]. Also, free-form deformation operates on raw pixel intensities, therefore illumination variations can be problematic. Features such as the orientation histogram or the average motion are robust to registration errors to an extent. The representation features are computationally simple, however, free-form deformation is computed through an iterative process which can keep the representation from achieving real-time processing speed.

2.5.5 Temporal bag-of-words representation

The temporal Bag-of-Words (BoW) representation is specific to AU detection [210] and can be best explained by describing how the problem is formulated by its authors. Simon *et al.* [210] assume that an AU is an event that *exists* in a given image sequence. The problem is then formulated as identifying the boundaries of the existing AU event in the given sequence. The approach was also generalized for multiple AUs [210].

Temporal BoW represents an arbitrary subset of the given image sequence with a single histogram which is computed as follows (Fig. 2.4f): 1) Each frame in the subset is represented using the part-based SIFT representation (Section 2.4.5) and compressed with Principal Component Analysis (PCA) to obtain a frame-wise vector, 2) each frame-wise vector is encoded using the BoW paradigm that measures similarity by means of multiple vectors via soft clustering [210], and 3) all encoded frame-wise vectors are collected in a histogram.

The sensitivity of the representation to illumination variations, registration errors, head-pose variations and identity bias is similar to the part-based SIFT representation. Unlike the part-based representation, temporal BoW does not encode componential information explicitly, as PCA can create holistic features (Section 2.6.3). Unlike other spatio-temporal representations, the temporal BoW does not encode temporal variation. The dimensionality depends on the size of the BoW vocabulary. The computational complexity of the representation mainly depends on the search performed on the visual vocabulary, particularly, the size of the vocabulary and the complexity of the search algorithm.

2.5.6 Deep Learning

Similarly to spatial representations (Section 2.4.6), an increasingly popular trend is to apply deep learning for learning spatio-temporal representations from data. A variety of deep learning architectures have been proposed. Liu *et al.* [124] proposed a deep architecture that learns deformable facial parts, namely, 3D Convolutional Neural Network Deformable Action Parts

(3DCNN-DAP). Combining appearance and shape features has also been a motivation while designing architectures, and there exist at least two studies that proposed such architectures: The Deep Temporal Appearance-Geometry Network (DTAGN) [93] and the Convolutional and Bi-directional Long Short-Term Memory Neural Networks [88]. Elaiwat *et al.* [59] proposed a restricted Boltzmann machine (RBM) network that is relatively 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. The concept of boosting-based learning has also been incorporated into deep architectures that were built for AU recognition [76, 126]. Another deep architecture is motivated by the fact that basic expressions can be decomposed into their AUs, therefore incorporates an "AU-aware" layer [123]. In addition to deep learning, other hierarchical representations are also proposed. Liu *et al.* [125] introduced 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.

Owing to their multi-layered structure, deep architectures can be robust to challenges that are present "in-the-wild", such as head-pose or illumination variations (see Section 2.4.6). However, compared to their spatial counterparts discussed in Section 2.4.6, spatio-temporal deep architectures have not been as prominent in the affect recognition competitions that are "in-the-wild" or those that include naturalistic affective behaviour with head-pose variations. Neither the winner [270] nor the runner-up [98] of the video-based sub-challenge of EmotiW'15 have relied on deep learning[1]. The third system has relied on recurrent neural networks for modelling temporal evolution, however this was a relatively shallow architecture [55]. Moreover, neither the winners of the FERA'15 occurance sub-challenge [272] and intensity sub-challenge [155], nor the runner up system of both challenges [15] relied on deep learning[2]. The third system in the occurrence sub-challenge used a deep architecture [72]; however, interestingly, this was not a spatio-temporal representation but a spatial one. In summary, there is room for improvement for spatio-temporal deep architectures in facial expression analysis. The ones that proved successful are relatively shallow [55, 59, 93], partly due to the overfitting risk that is caused by the limited size of the datasets available for video-based facial expression analysis.

[1]While the winning system [270] showed some performance improvement with the inclusion of one CNN, this improvement is relatively small and the overall structure of the system is not deep.

[2]The performances of the systems have been provided in `http://ibug.doc.ic.ac.uk/resources/FERA15/`, and the references of those systems in [88]. We have not considered in our analysis the systems that did not participate to the competition.

This thesis will also introduce a hierarchical learnt representation in Chapter 4, but one that has two layers, therefore would be called shallow in today's literature. The proposed representation will be validated on expressions of emotions. Among the above-listed representations, the ones that were similarly proposed for expressions of emotions [59, 93, 123–125] have generally outperformed engineered features in datasets with large (*i.e.* pronounced) facial activity, such as CK+ [96] or MMI [167]. However, a learnt spatio-temporal representation may have some generalisation disadvantages. The representation may be sensitive to the frame rate of the training sequences, or the temporal phases of the expressions; that is, if all training sequences start with a neutral face and finish at the apex of the expression, then the representation may require the test sequences to strictly follow this order. Those existing studies [59, 93, 123–125] have been validated only through within dataset validation (*i.e.* the representations are learnt and tested on the same dataset) where such inconsistencies are not present — further validation is needed to test whether the learnt features produce meaningful representations on test sequences with different frame rate or different (orders of) temporal expression phases. Also, 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 [125].) Finally, those representations are tested only in recognising pronounced expressions, even though one of the most significant advantage of spatio-temporal representations is their ability to recognise subtle expressions. Table 2.3 summarises the learnt representations we discussed in this paragraph. As a comparison, Table 2.3 also lists some 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 [173] or six basic expressions [281]). 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 [173, 258]. The method in the last row of Table 2.3, Facial Bases [192], is the representation that we developed as part of this thesis' work. The design of this method is appropriate to tackle temporal inconsistencies and to recognise subtle expressions as well as pronounced expressions. Facial Bases will be discussed throughout Chapter 4 and compared to other methods in Table 2.3.

Table 2.3: 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.

Ref.	Approach	Engineered	Learnt	Needs Training Labels	Adressed Temporal Inconsis-tencies	Validation by Cross-database Representation Learning	Validated on Pronounced Expressions	Validated on Subtle Expressions
[281]	LBP-TOP	✓		N/A	N/A	N/A	✓	✓
[258]	Gabor Motion Energy	✓		N/A	N/A	N/A	✓	✓
[93]	DTAGN		✓	✓			✓	
[125]	Expressionlets		✓	✓†			✓	
[59]	Spatio-temporal RBM		✓	✓			✓	
[124]	3DCNN-DAP		✓	✓			✓	
[192]	Facial bases		✓		✓	✓	✓	✓

2.5.7 Discussion

The main motivation for spatio-temporal representations is to encode temporal variation in order to facilitate the recognition of subtle expressions [6]. Most systems used spatio-temporal representations with relatively simple registration strategies such as rigid registration based on 2 points. Relying on such simple registration, however, defeats the purpose of monitoring temporal variation, as the texture variation due to registration inconsistencies may be more evident than the variation due to facial activity. Although the free-form deformation representation addresses registration consistency through its own registration layer, the representation may fail in naturalistic settings (Section 2.5.4).

To address the demands of the spatio-temporal representations, Jiang *et al.* [92] detect a bounding box for the facial region in the first frame, and use this as a reference to register subsequent frames via Robust FFT. However, this pipeline overlooks two important factors. Firstly, although a finer registration may be achieved at the spatial level, this pipeline still maintains a frame-by-frame operation and does not address temporal consistency. Secondly, the subject may display large head-pose variations throughout the sequence, in which cases registration to a frontal face may result in failure. The registration demands that are not addressed in the current literature may have drawn the attention away from spatio-temporal representations in real world problems. This issue is also highlighted by the organisers of AVEC'13 [244] who despite arguing for the spatio-temporal LPQ-TOP representations' appropriateness, end up using LPQ due to the

challenging registration needs of LPQ-TOP.

A recent trend is to learn high-level spatio-temporal representations from data. Even though such representations may achieve high performance, the training sequences may impose restrictions on the generalisation ability of the representations. Learnt representations require further evaluation for their ability to produce meaningful representations when there are mismatches between the training and test data in terms of frame rate or (order of) temporal phases (Section 2.5.6).

2.6 Dimensionality Reduction

Dimensionality reduction can be used to address several affect recognition challenges such as illumination variation, registration errors and identity bias. Components that reduce dimensionality may operate across multiple layers, such as early preprocessing (*e.g.* downsampling input image, applying masks) and intrinsic representation layers. In this section, we group the additional dimensionality reduction techniques that follow the facial representation into three classes, namely pooling, feature selection and feature extraction methods.

2.6.1 Pooling

Pooling, a paradigm defined specifically for appearance representations, reduces dimensionality over local blocks of the representation by describing the features within the blocks jointly. This description discards the location of adjacent features and thereby increases the tolerance against registration errors. Such functionalities of pooling have a biological motivation as they mimic parts of mammals' vision systems [86, 174].

Pooling is usually applied on multiple small neighbourhoods across the image. There exists a variety of pooling techniques, such as binning features over local histograms, sampling the minimum or maximum value within a neighbourhood or computing the sum or average of the features across the neighbourhood [21, 22, 109]. Sensitivity to illumination variations is generally addressed by normalising the output of pooling (*e.g.* subtracting the local mean [174], or performing unit-norm normalisation [45]). Although pooling is mostly applied on the spatial domain, a number of studies apply pooling on spatio-temporal neighbourhoods as well (*e.g.* [127,219,258]).

Pooling is usually considered as an intrinsic layer of the representation [111]. Representations such as the low-level histogram representations (Section 2.4.2) are defined to be dependent

exclusively on a certain type of pooling (*i.e.* histograms). For these representations, we consider pooling as an intrinsic layer. The Gabor representations (Section 2.4.3) and spatio-temporal convolution with smooth filters (Section 2.5.3) have been used with a variety of pooling techniques as well as alternative dimensionality reduction schemes.

2.6.2 Feature selection

Feature selection aims at refining the facial representation by selecting a subset of its features, and optionally weighting the selected features. This process may be designed to have a semantic interpretation, such as discovering spatial [205, 266, 268, 285] or spatio-temporal [101, 282] regions of interest. Such applications of feature selection may reduce identity bias, as they are expected to discover the regions that are informative in terms of expressions rather than identity. Alternatively, the feature selection process may be designed to reduce dimensionality in a rather straightforward manner, without emphasis on the physical correspondence of the selected features [16, 91, 237].

Feature selection can be performed with a range of techniques. A simple form is selecting and weighting certain spatial regions manually [205]. Most systems rely on data-driven feature selection and the most popular paradigm is boosting. Boosting refers to a set of generic techniques, which are designed for prediction (classification/regression) [63]. Many affect recognisers neglect the prediction role of boosting techniques and use them only for feature selection. AdaBoost and GentleBoost [63] are the most widely employed boosting techniques. In addition to generic feature selection techniques, approaches tailored to affect recognition are also developed, for example to learn informative spatial regions by observing the temporal evolution of expressions [116].

The above-listed methods are supervised. One question while training supervised feature selectors is how the label information will be utilised. These techniques select features according to a two-class separation criterion (positive vs. negative). However, training datasets often include more than two classes. A common practice is to learn features separately for each class and group data as one-versus-rest (*e.g.* [91, 101, 116, 205]). Alternatively, features may be selected to facilitate the separation of all class pairs independently, *i.e.* one-versus-one training. Such feature selection schemes may be more useful, particularly for discriminating similar-looking expressions of different classes such as sadness and anger [282].

2.6.3 Feature extraction

Feature extraction methods extract novel features (*e.g.* holistic features) from the initial representations. They map an input representation onto a lower dimensional space to discover a latent structure from the representation. This transformation can be non-adaptive or adaptive (learnt from training data).

The most popular non-adaptive transformation is the Discrete Cosine Transformation (PCA) whereas the most popular adaptive transformation is Principal Component Analysis (PCA). PCA computes a linear transformation that aims at extracting decorrelated features out of possibly correlated features. Under controlled head-pose and imaging conditions, these features capture the statistical structure of expressions efficiently [26]. PCA is used by many systems including the winner of the AVEC continuous challenge [156].

A supervised alternative to the unsupervised PCA is Linear Discriminant Analysis (LDA). LDA uses label information to learn how to discriminate between differently labelled representations, and group similarly labelled representations. LDA can handle more than two classes as it considers only whether two arbitrary samples have the same or different labels. Most affect recognition systems train LDA using multiple classes simultaneously [23, 117, 157]. Alternative training schemes are also proposed. Kyperountas *et al.* [104] proposed a scheme where multiple LDA models are involved, and each model discriminates between a pair of classes.

The above-listed linear transformations are often used with representations that model the whole face [94, 129, 156, 241]. In such cases, they may render the overall pipeline susceptible to partial occlusions [226], as these transformations encode holistic information [185, 229].

Unsupervised [32, 136, 176] or supervised [206, 283] non-linear feature selection techniques are less popular than linear techniques. Shan *et al.* [207] showed that supervised techniques are usually more useful than unsupervised techniques. There is no strong evidence on the superiority of linear over non-linear feature extraction, or vice versa [207].

2.6.4 Discussion

The dimensionality of representations is often exploited to move representations to a higher level by discovering the spatial or spatio-temporal regions of interest, or selecting/extracting features that enhance the discrimination of similar-looking expressions of different emotions. To these ends, the vast majority of existing systems rely on generic dimensionality reduction techniques.

The optimality of such techniques, however, is being questioned in the scope of affect recognition, and new trends address the importance of making use of domain knowledge explicitly when developing dimensionality reduction techniques [252, 285].

2.7 Recognition

While the typical output of affect recognition systems is the label of an emotion or facial action, recent studies provide also the intensity of the displayed emotion or facial action [31, 74, 89, 94, 136, 186, 196, 244]. For AU recognition, the output can be enhanced significantly by providing the temporal phase of the displayed AU [101, 237, 242]. Also, to render the output more suitable to spontaneous behaviour, several studies recognise combinations of AUs [137, 224] rather than individual AUs as spontaneously displayed AUs rarely appear in isolation.

Except from a small number of unsupervised knowledge-driven approaches [115, 166], all affect recognisers use machine learning techniques. As any machine learning application, the performance of an affect recognition system depends on the quality and quantity of training data as well as the selected machine learning model.

2.7.1 Data

Labelling data is a challenging and laborious task, particularly for spontaneously displayed expressions and emotions. The annotation of spontaneously displayed emotions is challenging mainly due to the subjective perception of emotions [144], which is often addressed by using multiple annotators. However, combining multiple annotations is a challenge of its own [144]. Also, when annotation is carried out over sequences, there usually exists a delay between the perception and annotation of the annotator, which needs to be considered when combining the annotations. Recent attempts consider these issues and develop statistical methodologies that aim at obtaining reliable labels [144, 154].

Spontaneous AUs require frame-by-frame annotation by experts, and unlike posed AUs, where the subjects are instructed to display a particular (usually single) AU, the annotator has to deal with an unknown facial action which may be a combination of AUs [224]. A number of studies addressed the challenges in AU annotation and developed systems to assist annotators. De la Torre *et al.* [46] proposed a system that increases the speed of AU annotation with temporal phases, mainly by automating the annotation of onset and offset. Zhang *et al.* [278] developed an

interactive labelling system that aims at minimising human intervention and updates itself based on its own errors.

2.7.2 Statistical modelling

Most affect recognition systems rely on generic models such as Support Vector Machines (SVMs). Affect recognition has its own specific dynamics and recent studies aimed at tailoring statistical models for affect recognition. The new models address several issues such as modelling the temporal variations of emotions or expressions, personalising existing models, modelling statistical dependencies between expressions or utilising domain knowledge by exploiting correlations among affect dimensions.

Temporality — Modelling the temporal variation of facial actions or emotions proved useful [153, 237]. Typically used models are Hidden Markov Models (HMMs), which have been combined with SVM [237] or Boosting [101] to enhance prediction. Also, various statistical models such as Dynamic Bayesian Network (DBN) [224], Relevance Vector Machine (RVM) [153] or Conditional Random Fields (CRF) [14] are developed to learn temporal dependencies. Temporal variation is often modelled by systems that recognise the temporal phases of AUs [101, 237].

Personalisation — Identity cues render the generalisation of classifiers/regressors challenging. To deal with this, Chu *et al.* [34] proposed a method that can be used in conjunction with available discriminative classifiers such as SVM. The technique adapts the training data to a test sample by re-weighting the training samples based on the test subjects' identity cues.

Statistical Expression Dependencies — Facial activity is limited by face configuration and muscular limitations. Some facial actions cannot be displayed simultaneously, whereas some tend to co-occur. A number of AU recognition systems improve performance by exploiting these dependencies through statistical models such as DBNs [224, 225] or restricted Boltzmann machines [252].

Correlated Affect Dimensions — Although ignored by most dimensional affect recognisers, affect dimensions such as valence and arousal are intercorrelated [73]. Studies that extended RVM [153] and CRF [14] showed that modelling the correlation among affect dimensions may improve performance.

2.7.3 Discussion

The research efforts on creating affect-specific models (Section 2.7.2) are promising for affect recognition. However, to enable these models to focus on high-level semantics such as the temporal dependencies among AUs or inter-correlations between affect dimensions, the representations provided to the models must enable generalisation — the effects of illumination variations, registration errors, head-pose variations, occlusions and identity bias must be eliminated.

One way to provide informative features may be cascading two statistical models. For instance, the output of multiple SVM [237] or Boosting-based classifiers [101, 224, 225] may be passed to HMMs [101, 237] or DBNs [224, 225]. In such approaches, however, the first statistical model still suffers from challenges such as illumination variations unless they are addressed explicitly at representation level.

2.8 Summary

In this chapter we analysed facial affect recognition systems by breaking them down into their fundamental components and we highlighted their potentials and limitations.

The appearance representations that extract local features or involve local filtering are robust against *illumination variations* to an extent. Moreover, performing illumination normalisation at pooling (Section 2.6.1) can reduce the effect of illumination further. Illumination variations can be problematic for high-level representations that are extracted from raw pixel values. Such operations are common not only for the low-level histogram representations (see Section 2.4.2) and the Gabor representation (Section 2.4.3), but also for higher-level hierarchical representations (see Section 2.4.6). Shape representations are not affected by illumination as they ignore pixel intensities. However, (point) registration accuracy can decrease with illumination variations, thus degrading the performance of shape representations.

There has been significant progress in landmark localisation in recent years, and there exist multiple techniques that are robust to head-pose variations, occlusions and to illumination variations (see Section 2.3.5). Moreover, appearance representations are robust against small *registration errors* due to pooling or usage of smooth filters. Therefore, registration can be considered as a solved problem for affect analysis pipelines that utilise static appearance representations. Registration errors are problematic for shape representations (Section 2.4.7). Also, spatio-temporal representations that encode temporal variation suffer from registration errors as

they may interpret temporal registration errors as facial activity (Section 2.5.7).

Most representations encode componential features and deal with *occlusions* to an extent as the features extracted from unoccluded regions remain unaffected — a number of studies measured performance in presence of occlusions explicitly [39, 85, 147, 284]. Yet, representing irrelevant information from occluded regions can be problematic for subsequent steps such as dimensionality reduction (Section 2.6.3). Sparse representations can address occlusions more explicitly (Section 2.4.4). Another approach can be detecting occluded regions and removing them from the representation [85].

Head-pose variations remain mostly unaddressed at representation level. Part-based representations can address the problem partially (Section 2.3.5) and they have been preferred by some winning systems in affect recognition competitions with data that includes head-pose variations [155, 156, 272]. An alternative solution to dealing with head-pose variations is to learn the relationship between head-pose and expression variation at recognition level through statistical modelling [224], however, this approach may impose a large burden on the recognition process. Deep architectures trained with large amounts of data can also address head-pose variations successfully (Section 2.4.6). It must be noted that suppressing head-pose variations to favour the analysis of (non-rigid) facial motions is not the only way to facial affect analysis, as recent studies show that head-pose variation itself is a useful indicator of affective state (Section 2.3.5).

Identity bias is problematic for the popular low-level representations, which are adapted straightforwardly from face recognition. The majority of affect recognition systems do not address identity bias, but those that address it benefit from doing so. For example, in FERA'11 emotion challenge, the winner was the only system that considered identity bias through avatar image registration [269]. Similarly, the runner up of the FERA'15 intensity sub-challenge and AU occurrence sub-challenge addressed identity bias explicitly [15]. Indeed, when it comes to distinguishing the intensity of facial actions addressing identity bias can be particularly important [89]. Several representations address identity bias subject to the availability of the neutral face, which is a strong assumption for real-life applications. Identity bias can be tackled further at recognition level by adding a personalisation component [34] to discriminative classifiers (Section 2.7.2).

Combining *shape and appearance* features is one of the strategies that proved useful for the past decade and keeps being relevant also for today's more sophisticated systems. The win-

ning systems in various affect recognition competitions combined shape and appearance features [15, 155, 156, 203]. The advantage of combining features is that it is a design principle that can be implemented in multiple ways; while earlier approaches relied on relatively simpler strategies (*e.g.* training multiple classifiers and performing decision-level fusion), recent approaches designed deep architectures that learn features from appearance and shape representations concurrently [88, 93]. Combining features in general is in accordance with the behaviour of the human vision system when dealing with particularly ambiguous facial displays [24, 280] or interpreting different types of expressions [4].

Deep learning had a high impact in facial affect analysis recently (Section 2.4.6), and the interest in exploring deep architectures is expected to grow even further. Methods that rely on deep learning enjoy operating "in-the-wild" thanks to their multi-layered structure. The success of deep learning is very visible for spatial representation pipelines; the winners of image-based "in-the-wild" competitions employ, by and large, deep learning (Section 2.4.6). However, the picture is not the same for spatio-temporal representations yet. The winners of recent competitions for affect analysis "in-the-wild" or in naturalistic conditions with head-pose variations seldom relied on deep learning (Section 2.5.6).

While data-driven spatio-temporal representations are likely to receive increasing attention, learning a facial representation from time-dependent data may impose restrictions that are not visible through standard within-database evaluation protocols (Section 2.5.6). Let us conclude this chapter with three remarks that can be useful while devising data-driven spatio-temporal representations for naturalistic facial expressions.

- *Systems must recognise both subtle and pronounced expressions.* While spatio-temporal representations offer an important capability for recognising subtle expressions, many of recent data-driven spatio-temporal representations are validated exclusively on pronounced expressions (see Section 2.5.6). In general, systems are tested either on subtle or pronounced expressions. From the perspective of a real-world application, it is more plausible to approach automatic facial affect recognition as a unified problem across expression intensities.

- *Cross-database validation is essential for learnt spatio-temporal representations.* Cross-database tests become more and more widespread for static representations, however, this is not generally the case for spatio-temporal representations. In fact, cross-database rep-

resentations are particularly important for the latter. Spatio-temporal representations can develop sensitivity to the frame rate, or to the order of observed temporal phases (*e.g.* neutral-onset-apex), particularly if all the training sequences contain strictly the same temporal phases. Expressions do not always follow a standard order of temporal phases (Section 1.3); moreover, in naturalistic affective interactions, some of the temporal phases may not be visible due to partial occlusions, motion blur or sudden changes in head-pose.

- *Depending on a single emotion model may limit the representation's applicability.* Unlike a typical machine learning problem, the label of samples in facial expression analysis varies based on the emotion model that is being used. Supervised learning based only on one specific model may limit the relevance of the learnt representation, particularly if the model itself is limited in its ability to represent daily-life emotions (*e.g.* the six-basic emotions, see Section 1.3). Also, the optimal representations for more comprehensive emotion models can be fundamentally different from those for the simpler models. For example, the continuous emotion models benefit from machine learning pipelines that capture the temporal correlations within sequences (see Section 2.7.2).

Chapter 3

Registration of facial sequences

3.1 Introduction

As we discussed in the previous chapter, spatio-temporal representations are useful particularly for recognising the subtle expressions that are prevalent in daily life, however, those representations require accurate sequence registration (see Section 2.3.4). This chapter presents a robust registration technique that can be used both for whole-face and for part-based sequence registration.

Rigid registration for facial analysis needs to address multiple challenges, namely *non-uniform illumination variations*, *occlusions* and *facial activity* itself, which generates non-rigid motions that become outliers for rigid registration. Moreover, significant *drift errors* may accumulate over time with online registration, even when individual registration errors remain under a tolerance threshold, thus leading to registration failures. Undetected *registration failures* then become false references for subsequent frames, thus generating additional registration errors.

Registration is often approached as an optimisation problem and solved with a gradient-descent method [12, 60, 148, 164, 233]. However, gradient descent may underperform with untextured regions, particularly when high-gradient regions are associated with outlier motions. An emerging approach to optimisation in computer vision is using statistical learning [51, 215, 231, 261]. The general idea is to construct an algorithm by learning the relationship between the parameters to be optimised and the error caused by non-optimal parameters [38]. We argue that optimisation based on learning is also promising for rigid facial registration, as invariance to non-

rigid motions can be improved by training with sequences that contain facial activity. Moreover, robustness to non-uniform illumination variations can be improved with a robust feature extraction scheme without analytically modelling the relationship between features and misalignment parameters.

The framework presented in this chapter uses optimisation via statistical learning for rigid facial registration. This iterative framework (Fig. 3.1) reduces drift errors by computing Gabor motion energy with respect to multiple reference frames and can identify and correct registration failures via probabilistic learning. We show that, in iterative registration, misalignment can be estimated effectively with a pre-trained regressor of Gabor motion energy and that this regressor can generalise and perform accurately on data with illumination variations even when trained using controlled data. Moreover, we show that the ℓ_2 norm of Gabor motion energy can be used to train multiple regressors with different granularities and also to efficiently perform coarse-to-fine registration with these regressors. We refer to the proposed framework as MU-MIE (Multiple regressors for Misalignment Estimation), and evaluate it both for whole-face and part-based registration and obtain significantly higher accuracy than classical registration frameworks. Particularly notable is the part-based registration performance of the proposed framework in the presence of large facial activity due to facial expressions, and its robustness to non-uniform illumination variations.

The rest of this chapter is organised as follows. Section 3.2 presents the problem formulation. Section 3.3 outlines the proposed registration scheme. Section 3.4 describes the computation of illumination-normalised Gabor motion energy that is used as input for the regressors. Section 3.5 describes how Gabor motion energy is converted into rigid misalignment parameters. Section 3.6 explains how registration failures are identified and corrected. Experimental results are discussed in Section 3.7. Section 3.9 concludes the chapter.

3.2 Problem formulation

Let $\mathbf{S} = (I_1, I_2, \ldots, I_t, \ldots, I_T)$ be a sequence of arbitrary length T with unregistered frames I_t. The goal is to generate a registered sequence $\bar{\mathbf{S}} = (\bar{I}_1, \bar{I}_2, \ldots, \bar{I}_T)$ with no rigid misalignment between any two frames $\bar{I}_j, \bar{I}_k$. When $\mathbf{S}$ is acquired via streaming, a frame I_t must be registered as soon as it is obtained (*online registration*), where I_1 is the reference frame that subsequent frames will be registered to (*i.e.* $\bar{I}_1 = I_1$).

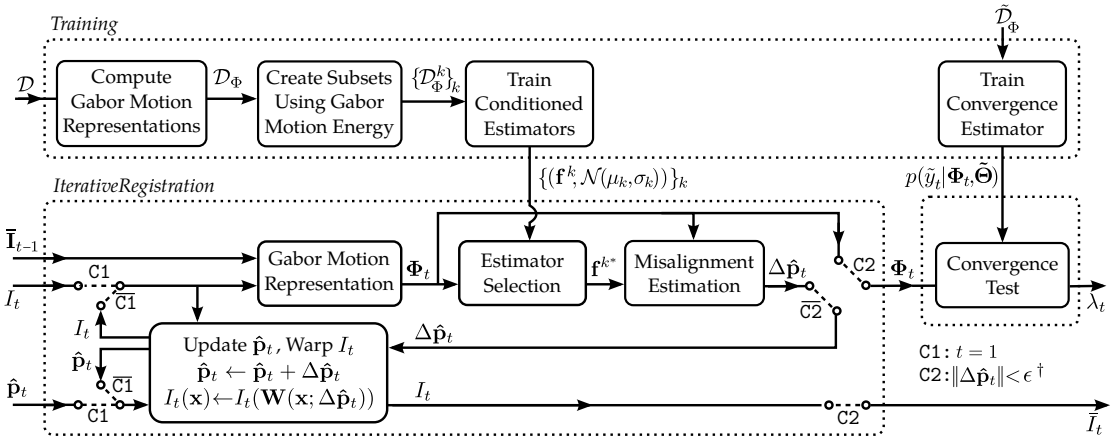


Figure 3.1: Overview of the proposed MUMIE framework. The top part represents the training of the misalignment estimators. The bottom part represents the iterative registration scheme, followed by a convergence test. The input to registration is an ordered set of reference frames $\bar{\mathbf{I}}_{t-1}$, the misaligned frame I_t and the initial misalignment estimation $\hat{\mathbf{p}}_t$. The dashed lines represent the conditional paths that are followed when the labelled conditions hold (C1/C2) or do not hold ($\bar{\text{C1}}/\bar{\text{C2}}$), and $||\cdot||$ is the ℓ_2 norm. †The condition C2 is satisfied also if a maximal number of iterations, $K_{\max}$, is reached.

Let $\mathbf{p}_t$ be the parameters of the rigid motion responsible for the misalignment in I_t. $\bar{I}_t$ can be obtained by transforming I_t with a warping operator $\mathbf{W}(\mathbf{x};\mathbf{p}_t)$ that maps each pixel $\mathbf{x} = (x,y)^T$ based on $\mathbf{p}_t$ [12]:

$$\bar{I}_t(\mathbf{W}(\mathbf{x};\mathbf{p}_t)) = I_t(\mathbf{x}). \tag{3.1}$$

The critical task is to obtain an accurate estimation of rigid motion, $\hat{\mathbf{p}}_t$. The rigid motion in I_t can be estimated with respect to a single frame (for example the most recently registered frame, $\bar{I}_{t-1}$); or by considering multiple past reference frames. For example, one can use an ordered set that contains the last T_R registered frames $\bar{\mathbf{I}}_{t-1} = (\bar{I}_\tau, \bar{I}_{\tau+1}, \ldots, \bar{I}_{t-1})$ where $\tau = \max\{1, t - T_R\}$. We refer to registration with $T_R = 1$ as *single-frame* registration and $T_R > 1$ as *multi-frame* registration.

3.3 Registration via learning

Faces are non-planar objects and compensating for rigid motion with an affine or projective transformation may distort facial geometry and undermine facial activity analysis. Therefore, we model rigid motion as a Euclidean transformation.

Let $\mathbf{p}_t$ be $\mathbf{p}_t = (p_1, p_2, p_3, p_4)$, where the elements define the horizontal and vertical translation, scale and rotation, respectively. Registration via optimisation starts computing the rigid

Figure 3.2: Illustration of drift errors that can occur over time, through an exemplar sequence that starts and ends with the same eye expression. Registration output of a Lucas-Kanade (LK) method [233] (top) and MUMIE (bottom). LK is prone to drift errors, as seen by comparing the first and last frames of the registered sequences. Drift errors are highlighted in the last column where the difference between the first and last frames is depicted. (Dark values indicate registration errors.)

motion between two images $\bar{I}_{t-1}, I_t$ with an initial estimate $\hat{\mathbf{p}}_t$ that is then updated iteratively as:

$$\hat{\mathbf{p}}_t \leftarrow \hat{\mathbf{p}}_t + \Delta\hat{\mathbf{p}}_t, \tag{3.2}$$

until the norm of the increment, $||\Delta\hat{\mathbf{p}}_t||$, is smaller than a threshold ε. $\Delta\hat{\mathbf{p}}_t$ is often computed with variations of the LK algorithm [148, 164], which use gradient descent for optimisation. Convergence is successful under constant illumination conditions and limited occlusions [12]. Extensions of LK can tackle illumination variations and occlusions using a robust estimator [13] or a cosine kernel that eliminates outliers caused by local texture mismatches [233]. However, algorithms based on gradient-descent may underperform when high-gradient image regions are related to outlier motions.

Registration for facial analysis needs to cope with the *non-rigid motions caused by facial activity*, which affect a large proportion of pixels and are problematic for part-based registration. Facial activity evolves slowly and may not be eliminated as a local mismatch, thus causing drift errors. Fig. 3.2 illustrates this problem: the first and last frame should be aligned as they depict the same eye expression at two different instants of a sequence; however, another expression appearing in the in-between frames causes drift errors for the LK-based algorithm [233].

An emerging approach to optimisation is to perform the updates with a pre-computed function [38,231,261]. The increment $\Delta\hat{\mathbf{p}}_t$ can be computed with a regressor that models the relationship between misalignment and the errors caused by misalignment. We use regression for rigid facial registration. At each iteration, we compute the rigid motion between $\bar{I}_{t-1}$ and I_t (or more generally, between $\bar{\mathbf{I}}_{t-1}$ and I_t) with a regressor $\mathbf{f}$ as:

$$\Delta\hat{\mathbf{p}}_t = \mathbf{f}(\Phi(\bar{\mathbf{I}}_{t-1}, I_t); \Theta), \tag{3.3}$$

where Θ is the vector of input-independent regressor parameters, and $\Phi(\cdot)$ is a feature extraction process that we discuss later in this section. Θ is computed from a dataset $\mathcal{D} = \{(\bar{\mathbf{I}}^n, I^n, \mathbf{p}^n)\}_{n=1}^N$ that contains N misaligned samples and their misalignment labels. Invariance against an outlier can be encouraged by augmenting $\mathcal{D}$ with training samples that are affected by the outlier [19]. Moreover, invariance against illumination variations can be encouraged with a robust feature extraction scheme. Unlike algorithms based on gradient descent (*e.g.* LK), the computation in (3.3) does not require a differentiable expression to minimise. For this reason, we can employ feature extraction schemes that are difficult to differentiate or not differentiable. However, while optimisation with regression provides an efficient means for dealing with outliers, an important issue has to be addressed for a pre-computed regressor, namely the *generalisation to unseen faces and imaging conditions*.

3.4 Encoding local motion with speed- and orientation-selective filters

Generalisation can be improved with a feature extraction scheme that is sensitive to rigid motion and insensitive to irrelevant factors, such as skin colour and illumination variations. To this end, we use a spatio-temporal Gabor representation, which encodes motion without computing motion vectors explicitly [2] and is robust against illumination variations. The Gabor representation encodes the motion between two frames $\bar{I}_{t-1}$ and I_t by convolving this pair with speed- and orientation-selective Gabor filters that are defined as [172]

$$g_{v,\theta}^{\phi_{\text{off}}}(x,y,t') = \frac{\gamma}{\sqrt{8\pi^3\sigma^2\tau}} e^{-\frac{\bar{x}^2+\gamma\bar{y}^2}{2\sigma^2} - \frac{(t-\mu_t)^2}{2\tau^2}} \cos\frac{2\pi}{\lambda}(\bar{x} + vt' + \phi_{\text{off}}) \tag{3.4}$$

where $\bar{x} = x\cos\theta + y\sin\theta$ and $\bar{y} = -x\sin\theta + y\cos\theta$, and the phase offset ϕ_{off} can be set to $\phi_{\text{off}} = 0$ to obtain an even-phased (cosine) filter and to $\phi_{\text{off}} = \frac{\pi}{2}$ to obtain an odd-phased (sine) filter — the two filters together form a quadrature pair. The parameters θ and v define the orientation and speed of motion that the filter is tuned for (see [172] for the definition and details of the remaining parameters). An important property of the Gabor representation is direction selectivity (*e.g.* distinguishing between leftwards and rightwards motion), which is acquired by computing the *Gabor motion energy* through a quadrature filter pair as [2]:

$$E_{\mathbf{I}_t} = (\mathbf{I}_t * g_{v,\theta}^{0})^2 + (\mathbf{I}_t * g_{v,\theta}^{\frac{\pi}{2}})^2, \tag{3.5}$$

where $*$ denotes convolution and $\mathbf{I}_t$ is defined as $\mathbf{I}_t = (\bar{I}_{t-1}, I_t)$.

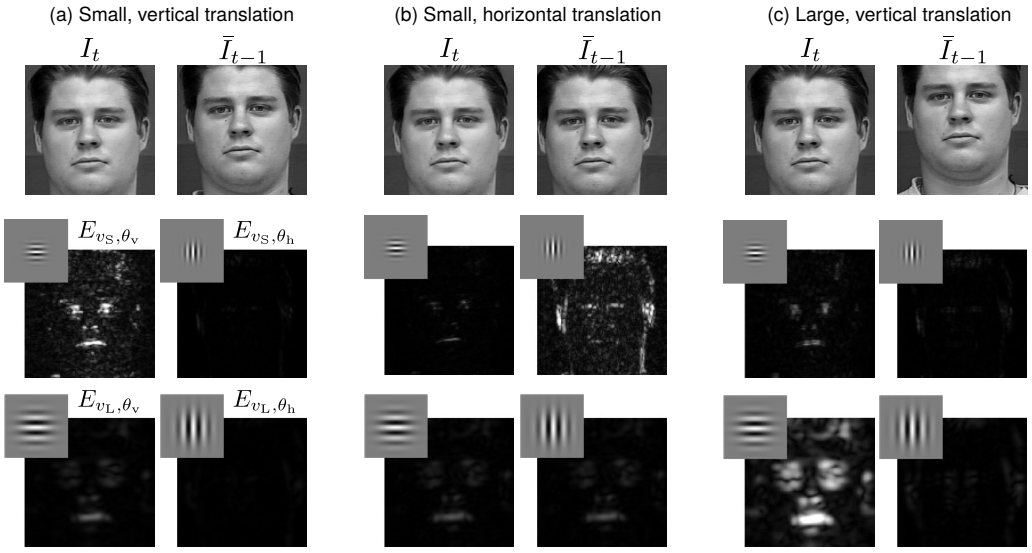


Figure 3.3: Illustration of the usefulness of the Gabor motion energy for registration via three example cases (a–c) that involve different types (horizontal/vertical translation) and amounts (small/large) of misalignment. For each case, the Gabor motion energy is computed with four different filter pairs tuned to a particular speed ($v_{\text{S(mall)}}$ or $v_{\text{L(arge)}}$) and orientation ($\theta_{\text{h(orizontal)}}$ or $\theta_{\text{v(ertical)}}$). The energy always becomes maximal when the filters are in tune with the misalignment.

Fig. 3.3 illustrates why the Gabor representation is useful for registration. We plot three pairs of images along with the motion energies computed through four pairs of Gabor filters. The energy produced with filters tuned to small, vertical motions gets maximal when the misalignment involves a small, vertical translation, as illustrated in Fig. 3.3a. More generally, misalignments in different directions or magnitudes activate different Gabor filters. This property is critical for optimisation, as it implicitly guides which direction each optimisation step should take, and what the step size should be. The usage of such a motion representation enables generalisation; if we would replace the images $\bar{I}_{t-1}$, I_t in Fig. 3.3 with the images of other subjects, the energy output would change, however, the essential relationship would not: each filter would still reach its maximal response only if the rigid motion (*i.e.* the misalignment) is in tune with the filter parameters.

3.4.1 Motion energy of a moving line

The motion energy, which models the lower layers of the visual cortex, is typically analysed based on the response of a visual cell to a moving line [2]. We also analyse motion energy over an exemplar moving line, as the moving line has a closed-form mathematical expression and therefore its response can be studied analytically. The closed-form expression of a moving line

has been computed for a line with one (spatial) dimension [177], however, to the best of our knowledge, it has not been computed for a line with two spatial dimensions. In this section we first obtain the closed-form expression of Gabor motion energy for a (two-dimensional) moving line, and then show how to tune a Gabor filter to a particular speed and direction. We will then discuss how to reduce the illumination sensitivity of the Gabor motion energy, and then return to its usage for estimating rigid misalignment.

Let $\mathbf{I}_l \triangleq \mathbf{I}_l(x,y,t)$ denote a sequence of a moving line as:

$$\mathbf{I}_l(x,y,t) \triangleq c\delta\left(x\cos\theta_l - y\sin\theta_l - tv_l\right), \tag{3.6}$$

where δ is Dirac's delta, x,y,t denote spatial coordinates and time; θ_l defines the orientation of $\mathbf{I}_l$; $v_l \geq 0$ defines the speed and $c > 0$ is the luminance value of $\mathbf{I}_l$. The computation of the line's energy, $E_{\mathbf{I}_l} = (\mathbf{I}_l * g^0_{v,\theta})^2 + (\mathbf{I}_l * g^{\frac{\pi}{2}}_{v,\theta})^2$, is difficult due to the triple integrals involved in the convolutions. With the help of the Convolution Theorem, we compute the $E_{\mathbf{I}_l}$ using Mathematica® as (see Appendix A.1):

$$E_{\mathbf{I}_l} = \frac{\bar{c}^2}{1+v_l^2} e^{-\frac{v_g v_l \cos\theta_{gl} + 4tyv_l \sin\theta_l - 4x\cos\theta_l(tv_l + y\sin\theta_l)}{1+v_l^2}} e^{-\frac{1+4x^2+4y^2+2v_g^2+(2+8t^2)v_l^2-\cos 2\theta_{gl}+4(x^2-y^2)\cos 2\theta_l}{4(1+v_l^2)}}. \tag{3.7}$$

Then, using the second derivative test, it can be shown that in order to tune the filter pair to a line moving with spatial orientation θ_l and speed v_l, the filter parameters v_g and θ_g must be defined as follows (see Appendix A.2):

$$v_g = v_l, \tag{3.8}$$

$$\theta_g = \pi - \theta_l + 2\pi k. \tag{3.9}$$

We can obtain a complete motion representation by computing multiple energy functions, each involving a different filter pair tuned to a different speed and orientation [2]. Such a representation enables the identification of the speed and direction of an *unknown* line: the Gabor filters that are tuned to a motion similar to that of the unknown line would produce a higher energy than other filters. In Fig. 3.4a we show how the motion energy computed from multiple Gabor filters enables us to identify the orientations of two different lines (one at each row) that move with the same speed. The line at the top of Fig. 3.4a moves with an orientation of $\frac{\pi}{2}$ and the maximal energy is produced with the spatio-temporal Gabor filter that is tuned to the speed of the line,

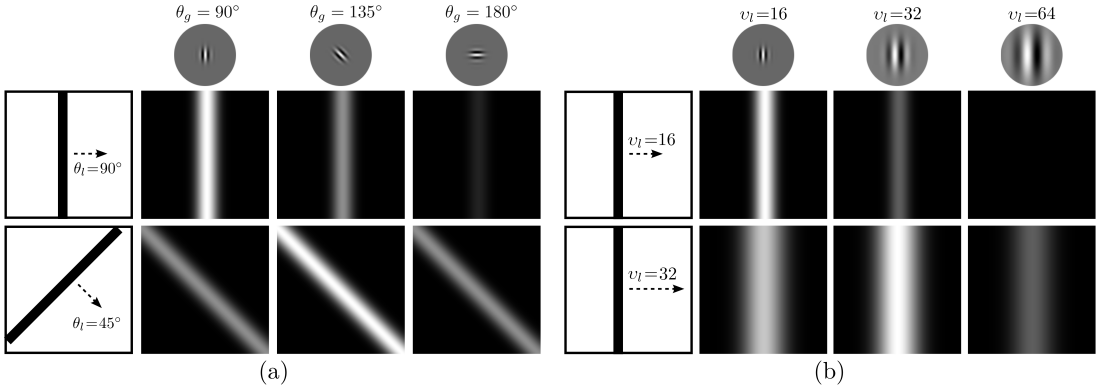


Figure 3.4: Two exemplar cases that illustrate how Gabor motion energies computed from multiple pairs of spatio-temporal Gabor filters enable the identification of motion speed and orientation. (a) Two lines that are moving with the same speed but in different orientations (90° and 45°): The maximal energy for each line is produced with the filter pair that is tuned to the lines' orientation. (b) Two lines that are moving in towards the same direction but with different speeds (16 and 32): The maximal motion energy is produced with the filter that is tuned to the lines' speed.

that is, according to (3.9), $\theta_g = \frac{\pi}{2}$. Similarly, the maximal energy for the line at the bottom of Fig. 3.4a is produced by the filter with $\theta_g = \frac{3\pi}{4}$ as this is the filter that is tuned to the orientation of the line (*i.e.* $\frac{\pi}{4}$). A similar discussion applies for the examples in Fig. 3.4b, where we show how energy can be used to discover the speed of two lines: The maximal motion energies are produced by the filters that are tuned through (3.8) to the speed of each line.

3.4.2 Contrast normalisation for motion energy

The motion energy is sensitive to the brightness of the moving elements; this can be seen in (3.7) where the energy is proportional to the brightness-related coefficient, c^2. Brightness/illumination sensitivity is an undesired property for a motion representation. In this section we propose a normalisation scheme to reduce illumination sensitivity. We show how this scheme eliminates the dependence on the initial brightness of a moving line and we extend it to tackle temporal illumination variations for generic sequences.

Normalisation of Line Brightness

We aim to obtain a *normalised sequence* $\tilde{\mathbf{I}}_l \triangleq \tilde{\mathbf{I}}_l(x,y,t)$ such that the energy of this sequence, $E_{\tilde{\mathbf{I}}_l}$, is illumination-independent and yet its functional form is still equal to that of $E_{\mathbf{I}_l}$. Such a sequence can be obtained by dividing the frames of $\mathbf{I}_l$ with a coefficient that is proportional to c^2; this is akin to the *contrast normalisation* that is arguably employed by the mammal visual cortex [79, 250]. We now show how such a coefficient can be obtained.

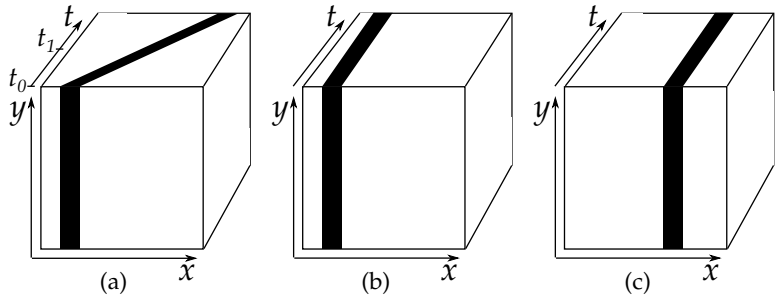


Figure 3.5: Illustration of how we create static sequences. (a) $\mathbf{I}$: sequence of a line that moves horizontally. (b) $\mathbf{I}^{t_0}$: static sequence created from $\mathbf{I}$ using the frame at time t_0; (c) $\mathbf{I}^{t_1}$: static sequence created from $\mathbf{I}$ with the frame at t_1.

Let us assume that we have a sequence of a *static* line whose luminance value is c. Then, according to (3.7), the energy of the static line will be proportional to c^2 and, because the line is not moving, the energy will be constant over time. The energy of this static line provides us with the coefficient that we need for normalisation: A coefficient that is constant over time and proportional to c^2.

In fact, we *do* have a way of obtaining such a sequence: We can take a frame from $\mathbf{I}_l$ at any time t_k, and obtain a static sequence, $\mathbf{I}_l^{t_k} \in \mathbb{R}^3$, by replicating this frame over time. We illustrate this for the exemplar horizontally moving line in Fig. 3.5a by creating two static sequences, $\mathbf{I}_l^{t_0}, \mathbf{I}_l^{t_1}$ (see Fig. 3.5b,c). Such static sequences obtained from a sequence $\mathbf{I}_l$ can be defined as $\mathbf{I}_l^{t_k}(x,y,t) \triangleq \mathbf{I}_l(x,y,t_k)$.

Let us obtain our normalisation coefficient using the static line at time $t_k = 0$. The speed of this static line, v_l, is zero, and therefore according to (3.7) its energy is:

$$E_{\mathbf{I}_l^0} = \bar{c}^2 e^{-\frac{1+4x^2+4y^2+2v_g^2+4(x^2-y^2)\cos 2\theta_l - \cos 2\theta_{gl} - 8xy\sin 2\theta_l}{4}}. \tag{3.10}$$

As expected, the energy of this static line is constant over time and proportional to $\bar{c}^2$ (and c^2). To complete our normalisation, we need to extract a single coefficient from the function $E_{\mathbf{I}_l^0}$. This can be achieved by integrating $E_{\mathbf{I}_l^0}$ over the entire sequence domain, $\Omega = X \times Y \times T$. Let $Z_{\mathbf{I}_l^t}$ be a function that computes the *normalisation coefficient* as:

$$Z_{\mathbf{I}_l^t} \triangleq \int_{\Omega} E_{\mathbf{I}_l^t}(\mathbf{x}')\mathrm{d}\mathbf{x}', \tag{3.11}$$

where $\mathbf{x}' = (x', y', t')$. Then, the normalisation coefficient of $E_{\mathbf{I}_l^0}$ can be computed as:

$$Z_{\mathbf{I}_l^0} = \int_{\Omega} E_{\mathbf{I}_l^0}(\mathbf{x}')\mathrm{d}\mathbf{x}' = \bar{c}^2 \int_{\Omega} \frac{E_{\mathbf{I}_l^0}(\mathbf{x}')}{\bar{c}^2}\mathrm{d}\mathbf{x}' = \frac{c^2}{8\pi^4}S, \tag{3.12}$$

where S denotes the output of a definite integral that is another constant but one that does not depend on the illumination of the line. Finally, we obtain the normalised sequence as:

$$\tilde{\mathbf{I}}_l = \frac{1}{\sqrt{Z_{\mathbf{I}_l^0}}}\mathbf{I}_l = \frac{2\sqrt{2}\pi^2}{\sqrt{S}}\delta(x\cos\theta_l - y\sin\theta_l - tv_l). \tag{3.13}$$

As desired, the energy of this line, $E_{\tilde{\mathbf{I}}_l}$, will be independent of c, and its functional form would be equal to that of $E_{\mathbf{I}_l}$.

Normalisation of Temporal Illumination Variations

The normalised sequence in (3.13) was obtained by dividing the sequence with a single coefficient. To tackle temporal variations in a generic sequence (*i.e.* beyond a moving line), we divide each frame of the sequence with a separate (time-dependent) coefficient $Z_{\mathbf{I}^t}$:

$$\tilde{\mathbf{I}}(x,y,t) = \frac{1}{\sqrt{Z_{\mathbf{I}^t}}}\mathbf{I}(x,y,t), \tag{3.14}$$

where $Z_{\mathbf{I}^t}$ is computed as in (3.11) but, because now it is computed from a generic sequence without a closed-form expression, $\mathbf{I}$, we can represent it only as a definite integral.

To show that this extension is able to tackle temporal illumination variations for sequences without closed-form expressions, we recast the problem of illumination normalisation as follows. Consider a sequence $\mathbf{I}_p \triangleq \mathbf{I}_p(x,y,t)$ where there are no illumination variations, and another sequence $\mathbf{I}_q$ that contains the same motion as $\mathbf{I}_p$ but is affected by a temporal variation such as $\mathbf{I}_q(x,y,t) \triangleq (\alpha t + \beta)\mathbf{I}_p(x,y,t)$. Our goal with illumination normalisation is to have normalised versions of these sequences that have equal energies, that is, $E_{\tilde{\mathbf{I}}_p} = E_{\tilde{\mathbf{I}}_q}$.

The energies of normalised sequences can be written as:

$$E_{\tilde{\mathbf{I}}_p} = \left[\int \frac{\mathbf{I}_p(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_p^w}}} g^e(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \frac{\mathbf{I}_p(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_p^w}}} g^o(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2, \tag{3.15}$$

$$E_{\tilde{\mathbf{I}}_q} = \left[\int \frac{\mathbf{I}_q(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_q^w}}} g^e(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \frac{\mathbf{I}_q(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_q^w}}} g^o(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2, \tag{3.16}$$

where $\mathbf{u} = (u,v,w)$ and $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{u}$. Note that $\mathbf{I}_q^w(x,y,t) = \mathbf{I}_q(x,y,w) = (\alpha w + \beta)\mathbf{I}_p(x,y,w) = (\alpha w + \beta)\mathbf{I}_p^w(x,y,t)$, and because the convolution involved in energy computation is a linear operator, we can compute $Z_{\mathbf{I}_q^w}$ as:

$$\begin{aligned} Z_{\mathbf{I}_q^w} &= \int_\Omega E_{\mathbf{I}_q^w}(\mathbf{x}')\mathrm{d}\mathbf{x}' = \int_\Omega (\alpha w + \beta)^2 E_{\mathbf{I}_p^w}(\mathbf{x}')\mathrm{d}\mathbf{x}' \\ &= (\alpha w + \beta)^2 \int_\Omega E_{\mathbf{I}_p^w}(\mathbf{x}')\mathrm{d}\mathbf{x}'. \end{aligned} \tag{3.17}$$

Therefore, we can rewrite (3.16) as:

$$E_{\tilde{\mathbf{I}}_q} = \left[\int \frac{(\alpha w+\beta)\mathbf{I}_p(\mathbf{u})}{(\alpha w+\beta)\sqrt{Z_{\mathbf{I}_p^w}}} g^e(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \frac{(\alpha w+\beta)\mathbf{I}_p(\mathbf{u})}{(\alpha w+\beta)\sqrt{Z_{\mathbf{I}_p^w}}} g^o(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2$$
$$= \left[\int \frac{\mathbf{I}_p(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_p^w}}} g^e(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2 + \left[\int \frac{\mathbf{I}_p(\mathbf{u})}{\sqrt{Z_{\mathbf{I}_p^w}}} g^o(\bar{\mathbf{x}})\mathrm{d}\mathbf{u}\right]^2, \tag{3.18}$$

As desired, the energies of the sequences $E_{\tilde{\mathbf{I}}_p}$ and $E_{\tilde{\mathbf{I}}_q}$ are equal, which can be seen by comparing (3.15) and (3.18).

The $Z_{\mathbf{I}_p^w}$ in (3.18) that remains after cancelling out the temporal illumination variations depends on time w. This may cause the trend of the energy function to change during normalisation, which is prohibitive, as the tuning of the filter parameters v_g, θ_g was based on the energy function to follow a specific trend. Fortunately, $Z_{\mathbf{I}_p^w}$ shows little sensitivity to time w (see Appendix A.3), and therefore the trends of the normalised and un-normalised energy functions are similar.

It must be noted that our normalisation scheme is most applicable when local slices of a sequence are processed and normalised independently from one another, particularly in the presence of non-uniform illumination variations. Local processing and illumination normalisation are also biologically plausible [174] and are employed by state-of-the-art spatial [111] and spatio-temporal [219] image processing pipelines. However, local normalisation can be computationally expensive, particularly if normalisation is computed on spatially overlapping portions of the input images. In Appendix A.4 we describe how the locally normalised energy can be computed efficiently through the summed area tables [42].

3.4.3 Pooling with respect to multiple frames

Now that we showed how to normalise illumination, let us return to the usage of Gabor motion energy for estimating the rigid motion between the pair of images in $\mathbf{I}_t = (\bar{I}_{t-1}, I_t)$. This corresponds to single-frame registration (see Section 3.2), but we will shortly discuss the corresponding multiframe extension.

The overall representation that we use to estimate rigid misalignment from a pair, $\Phi'(\bar{I}_{t-1}, I_t) = (\phi_{t,1}, \phi_{t,2}, \ldots, \phi_{t,d}, \ldots, \phi_{t,D})$, is computed by pooling the normalised energy matrices $E_{\tilde{\mathbf{I}}_t}$ after partitioning them into $M \times M$ non-overlapping subregions. An advantage of pooling is to facilitate generalisation in terms of image size. While the size of the energy matrices $E_{\tilde{\mathbf{I}}_t}$ depends on the size of the images $\bar{I}_{t-1}$ and I_t, after pooling we have $M \times M = M^2$ coefficients per energy matrix independently of image size. The dimensionality of the overall representation is $D = M^2 \times K_G$,

where K_G is the number of Gabor filter pairs, or, equivalently, the number of energy matrices. The implementation details of the representation are provided in Section 3.7.3.

To reduce drift errors, we can compute the overall representation with respect to multiple reference frames, $\bar{\mathbf{I}}_{t-1}$. We denote this *multi-frame motion representation* between I_t and $\bar{\mathbf{I}}_{t-1}$ as $\Phi_t = \Phi(\bar{\mathbf{I}}_{t-1}, I_t)$. Φ_t computes pair-wise motion representations between the misaligned frame and each of the reference frames in $\bar{\mathbf{I}}_{t-1}$, and then averages them over time:

$$\Phi_t = \Phi(\bar{\mathbf{I}}_{t-1}, I_t) = \frac{1}{t-\tau} \sum_{t'=\tau}^{t-1} \Phi'(\bar{I}_{t'}, I_t), \tag{3.19}$$

where $\tau = \max\{1, t - T_R\}$.

3.5 Mapping motion energy into misalignment parameters

Throughout Section 3.4 we discussed how to encode the local motion between the misaligned frame I_t and the reference frame(s) in $\mathbf{I}_t$. This section describes how to convert local motion to rigid motion parameters as proposed in (3.3). For brevity, we rewrite this equation as

$$\Delta\hat{\mathbf{p}}_t = \mathbf{f}(\Phi_t; \Theta) \tag{3.20}$$

and denote the training set for the regressor $\mathbf{f}$ as $\mathcal{D}_\Phi = \{(\Phi^n, \mathbf{p}^n) : \Phi^n = \Phi(\bar{\mathbf{I}}^n, I^n)\}_{n=1}^N$.

As Fig. 3.3 exemplifies, there is a non-linear relation between the Gabor representation (input) and rigid motion parameters (output). Therefore, it is reasonable to choose a non-linear regression function to model the intended input-output relationship. We choose $\mathbf{f}$ to be a single-hidden-layer neural network as it is a well-established non-linear regressor and one whose properties are well understood [138]. Then the parameter vector Θ includes the hidden-layer weights, output layer weights and biases [19]. The optimal parameters Θ^* are those that minimise the regularised mean squared error on $\mathcal{D}_\Phi$:

$$\Theta^* = \arg\min_{\Theta} \sum_{n=1}^{N} ||\mathbf{p}^n - \Delta\hat{\mathbf{p}}^n||^2 + \alpha||\Theta||^2, \tag{3.21}$$

where $\Delta\hat{\mathbf{p}}^n = \mathbf{f}(\Phi^n; \Theta)$ and $\alpha \in (0, 1]$ is the regularisation parameter defined during training through cross-validation.

The iterative process in Fig. 3.1 can achieve accurate registration if the errors of the estimator $\mathbf{f}$ get smaller as the amount of rigid motion in I_t gets smaller. However, the initial error in a given I_t may be high, therefore $\mathcal{D}$ must contain samples with both large and small misalignments. In

a dataset with such a broad range of input–output mapping, because of the bias/variance trade-off [64] the estimator may not be able to attain the desired level of accuracy. Although bias can be reduced by increasing model complexity, this would increase the variance of the estimator, thus increasing the risk of overfitting [64]. We address this problem with a coarse-to-fine misalignment estimation, as discussed next.

To improve the bias/variance trade-off, one can employ a coarse-to-fine cascade of K estimators $\{\mathbf{f}^k\}_{k=1}^K$ with *coarse* estimators tuned to large amounts of misalignment and *fine* estimators tuned to small amounts of misalignment (*e.g.* [286]). Such a cascade produces better bias/variance trade-offs as each estimator models an input-output mapping with a smaller range [64]. However, typical coarse-to-fine estimation schemes use all the estimators in the cascade, even when the initial registration is small and the finest estimator would suffice [106].

Coarse-to-fine estimation is more efficient when coarse estimators are used only if the initial registration error is large. However, this can be achieved only if we have a prior cue about the amount of misalignment in I_t. Interestingly, our spatio-temporal Gabor representation provides this cue: Large-magnitude motion activates Gabor filters with large spatial support [172], as was exemplified in Fig. 3.3. For this reason, the ℓ_2 norm (magnitude) of the representation,

$$\tilde{\rho}_t = ||\Phi_t|| = \sum_{d=1}^{D} \phi_{t,d}^2, \tag{3.22}$$

generally gets larger as rigid motion gets larger. Fig. 3.6 illustrates this relationship, which allows us to use the ℓ_2 norm of a representation as a prior on the amount of misalignment I_t.

We exploit magnitude while constructing the estimators of different granularities, $\{\mathbf{f}^k\}_{k=1}^K$. We choose all estimators $\mathbf{f}^k$ to have the same structure, therefore the estimators differ in their granularity due to the dataset they are trained with. Coarse estimators are trained with samples of larger magnitude and fine estimators with samples of smaller magnitude.

Let us denote the training dataset of each estimator as $\mathcal{D}_\Phi^k$, with $\bigcup_{k=1}^K \mathcal{D}_\Phi^k = \mathcal{D}_\Phi$. A simple way to create the sets $\{\mathcal{D}_\Phi^k\}$ is to first compute the magnitudes of all training samples, $\mathcal{D}_{\tilde{\rho}} = \{\tilde{\rho}^n : \tilde{\rho}^n = ||\Phi^n||,\ \forall(\Phi^n, \mathbf{p}^n) \in \mathcal{D}_\Phi\}$, and to partition the range of $[\min\{\mathcal{D}_{\tilde{\rho}}\}, \max\{\mathcal{D}_{\tilde{\rho}}\}]$ into K uniform intervals. However, this partitioning would be sensitive to the sample with maximal magnitude $\max\{\mathcal{D}_{\tilde{\rho}}\}$, as a large $\max\{\mathcal{D}_{\tilde{\rho}}\}$ value would affect all intervals. Instead, we allow for non-uniform lengths. To this end, we cluster the set $\mathcal{D}_{\tilde{\rho}}$ into K clusters by using a Gaussian Mixture Model. Each cluster is a distribution $\mathcal{N}(\tilde{\rho}|\mu_k, \sigma_k^2)$ where the variance σ_k^2 controls distribution width and is learnt from data. We create a subset $\mathcal{D}_\Phi^k$ by picking the samples that are close to

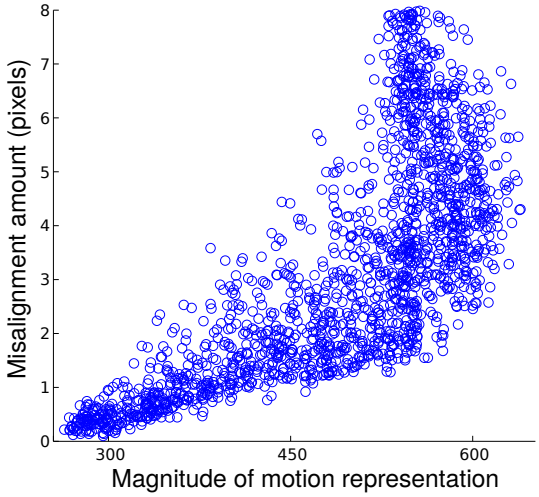


Figure 3.6: Illustration of the correlation between the magnitude of the Gabor representation and the amount of misalignment. This correlation suggests that the magnitude of the representation provides information about the amount of misalignment.

the k^{th} center. Specifically, we create $\mathcal{D}_\Phi^k$ as $\mathcal{D}_\Phi^k = \{(\Phi^n, \mathbf{p}^n) : \mathcal{N}(\tilde{\rho}^n | \mu_k, \sigma_k) \leq 2\sigma_k\}$ (i.e. we cover approximately 95% of the distribution with the $2\sigma_k$ rule [159]). Then, we train each $\mathbf{f}^k$ by applying the empirical risk minimisation in (3.21) using the dataset $\mathcal{D}_\Phi^k$.

We estimate misalignment at each iteration as:

$$\Delta\hat{\mathbf{p}}_t = \mathbf{f}^{k^*}(\Phi_t), \tag{3.23}$$

where $k^* = \arg_k \max \mathcal{N}(\tilde{\rho}_t | \mu_k, \sigma_k)$. For clarity, we dropped the regressor parameters.

If no registration failure occurs, the procedure described in this section can register a sequence $\mathbf{S}$ by registering each I_t sequentially for $t = 2, \ldots, T$ (Fig. 3.1). However, when the registration of a frame fails, the corresponding frame must be identified and removed prior to registering subsequent frames, otherwise it becomes a false reference for subsequent frames to register. This problem is addressed in the next section.

3.6 Failure identification and correction

To account for possible registration failures, it is desirable to generate a second output in addition to the registered sequence $\bar{\mathbf{S}}$. This second output, a vector λ, should indicate whether the registration at each frame was successful: $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_T)$, where $\lambda_t = 1$ indicates that $\bar{I}_t$ was registered correctly and $\lambda_t = 0$ indicates that registration failed.

Let $\langle \hat{\mathbf{p}}_t, \mathbf{p}_t \rangle_c$ define the average error in the estimation of canonical points [12] between two

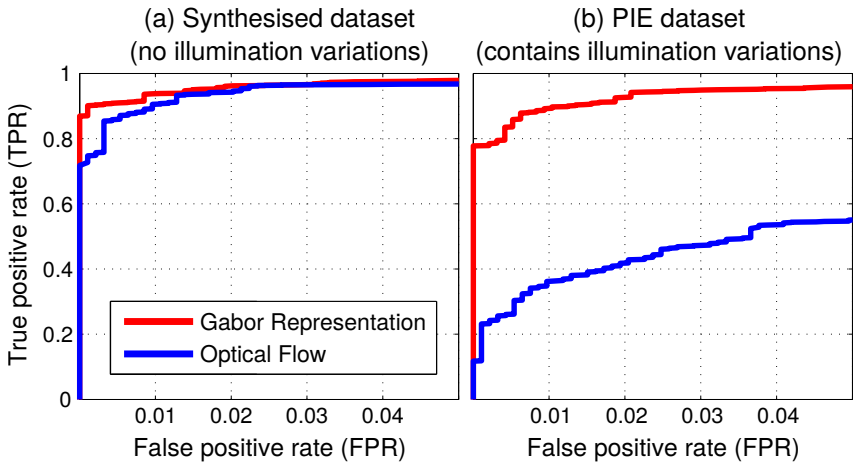


Figure 3.7: Failure identification performance on the Synthesised dataset (left) and on the PIE dataset (right) illustrated via ROC curves. The FPR range is restricted to $[0, 0.05]$ for better interpretation. Each curve is computed from 500 positive and 500 negative samples for $\varepsilon_y = 1$. Results suggest that the Gabor representation is more robust against illumination variations than the optical flow representation.

registered frames $\bar{I}_t$, $\bar{I}_{t-1}$. We choose two canonical points[1] $\mathbf{x}_1, \mathbf{x}_2$ as the leftmost and rightmost points in the vertical middle of the image plane. Then, $\langle \hat{\mathbf{p}}_t, \mathbf{p}_t \rangle_c$ can be computed as:

$$\langle \hat{\mathbf{p}}_t, \mathbf{p}_t \rangle_c = \sum_{i=1}^{2} \sqrt{||\mathbf{W}(\mathbf{x}_i; \hat{\mathbf{p}}_t) - \mathbf{W}(\mathbf{x}_i; \mathbf{p}_t)||}. \tag{3.24}$$

When this error is smaller than a *convergence threshold* ε_y, the registration is considered successful. We can cast failure identification as a binary classification problem where the two classes are *converged* (*i.e.* $\langle \hat{\mathbf{p}}_t, \mathbf{p}_t \rangle_c \leq \varepsilon_y$) and *not converged* (*i.e.* $\langle \hat{\mathbf{p}}_t, \mathbf{p}_t \rangle_c > \varepsilon_y$). For convenience we denote those two classes with a binary variable $\tilde{y} \in \{0, 1\}$.

This problem could be solved with a classifier trained with a labelled dataset $\tilde{\mathcal{D}}_\Phi = \{(\Phi_n, \tilde{y}_n)\}$. However, if we mislabel a frame $\bar{I}_t$ as *converged*, then the mislabelled frame will become a false reference to all subsequent frames; therefore, false positives are more costly than false negatives. A minimal false positive rate is therefore desirable, even if this causes a relatively higher rate of false negatives. False positives can be reduced if we have a confidence measure associated with each estimation, and we reject labelling a sample as *converged* unless the estimation confidence is above an *acceptance threshold* θ_{conv}. A probabilistic classifier can be used to this end, as the confidence value we seek is the probability assigned with the estimation.

We compute the convergence probability via Bayesian learning as [19]:

$$p(\tilde{y}_t = 1 | \Phi_t, \tilde{\mathcal{D}}_\Phi) = \int p(\tilde{y}_t = 1 | \Phi_t, \tilde{\Theta}) p(\tilde{\Theta} | \tilde{\mathcal{D}}_\Phi) \mathrm{d}\tilde{\Theta}, \tag{3.25}$$

[1] Two points suffice to define Euclidean motion.

where $\tilde{\Theta}$ is the vector of classifier parameters, $p(\tilde{\Theta}|\tilde{\mathcal{D}}_\Phi)$ is the prior distribution over parameters $\tilde{\Theta}$, and $p(\tilde{y}_t = 1|\Phi_t, \tilde{\Theta})$ is the probability of a segment with motion representation Φ_t having converged when the parameters are $\tilde{\Theta}$.

The closed-form expression of $p(\tilde{y}_t = 1|\Phi_t, \tilde{\Theta})$ depends on the classifier type, and the type of the distribution $p(\tilde{\Theta}|\tilde{\mathcal{D}}_\Phi)$ is usually selected in a way that would allow (3.25) to have a closed-form approximation (*i.e.* conjugate prior) [19]. Since the processes of failure identification and misalignment estimation share a common input space (*i.e.* spatio-temporal Gabor representation), we choose statistical models with the same structure, and use a single-hidden-layer neural network as a classifier. We implement Bayesian learning on this classifier through evidence approximation [149].

The decision on failure identification, λ_t, is defined as

$$\lambda_t = \lambda(\Phi_t) = \begin{cases} 1 & \text{if } p(\tilde{y}_t = 1|\Phi_t, \tilde{\mathcal{D}}_\Phi) > \theta_{\text{conv}} \\ 0 & \text{otherwise.} \end{cases} \tag{3.26}$$

We set the threshold θ_{conv} automatically as follows. We compute the ROC (receiver operating characteristic) curve of the failure identification function of λ by evaluating the true positive rate (TPR) and false positive rate (FPR) on a validation set for a range of threshold values θ_{conv}, and select the θ_{conv} that produces a low false positive rate (*e.g.* 0.01) on the ROC curve.

Fig. 3.7 illustrates the failure identification performance of the employed Bayesian neural network on two validation sets: one with constant illumination and one with illumination variations (the datasets are described in Section 3.7.2). To highlight the importance of a robust motion representation, we also compare the performance with an optical flow representation [61] that replaces the Gabor representation in the pipeline. Fig. 3.7a shows that a Bayesian neural network enables reliable failure identification with a TPR larger than 0.90 for a FPR as low as 0.01 for both representations. Fig. 3.7b shows that the Gabor representation is significantly more robust against illumination variations than the optical flow representation.

Let t_f denote a time when a registration failure occurs. This failure may be corrected by registering with respect to temporally farther frames. To this end, we search for a reference within a set of previously registered frames

$$\bar{\mathcal{I}} = \{\bar{I}_\tau : \bar{I}_\tau = \bar{I}_{\tau_{\min}}, \bar{I}_{\tau_{\min}+1}, \ldots, \bar{I}_{t_f-1} \wedge \lambda_\tau = 1\}, \tag{3.27}$$

where $\tau_{\min} = \max\{t_f - T_D, 1\}$ and T_D is the length of the temporal window within which correction is attempted. If a reference frame is found, then the failure is corrected and the registration

Algorithm 1 Procedure CORRECT

Input Failed frame, aligned frames: $(I_{t_f}, \bar{\mathcal{I}})$

Output Registered frame, convergence index: $(\bar{I}_{t_f}, \lambda_{t_f})$

for $\bar{I} \in \bar{\mathcal{I}}$ **do**
  $\bar{I}_{t_f} \leftarrow IterativeRegistration(\bar{I}, I_{t_f})$
  $\lambda_{t_f} \leftarrow \lambda(\Phi(\bar{I}, I_{t_f}))$
  if $\lambda_{t_f} = 1$ **then**
    break
  end if
end for
return $(\bar{I}_{t_f}, \lambda_{t_f})$

index is updated as $\lambda_{t_f} = 1$. The correction process is summarised in Algorithm 1. Note that *IterativeRegistration* refers to the set of operational blocks with the same label in the lower part of Fig. 3.1.

The likelihood of a correction can be increased by using frames after the failure time t_f, that is, by constructing $\bar{\mathcal{I}}$ as

$$\bar{\mathcal{I}} = \{\bar{I}_\tau : \bar{I}_\tau = \bar{I}_{\tau_{\min}}, \bar{I}_{\tau_{\min+1}}, \ldots, \bar{I}_{\tau_{\max}} \wedge \lambda_\tau = 1\}, \tag{3.28}$$

where $\tau_{\max} = \min\{T, t_f + T_D\}$. In this case, the registration process will have a delay of T_D frames. However, delays may become acceptable with small T_D values.

3.7 Experiments

In this section we validate the ability of the proposed framework to prevent drift errors, to perform robustly in the presence of facial expressions and non-uniform illumination variations, to identify failures reliably and to generalise to unseen conditions. We first compare multi-frame and single-frame registration for MUMIE. Then we compare MUMIE with state-of-the-art methods on sequences with facial expression variations and on sequences with non-uniform illumination variations; the latter cause registration failures, enabling us to evaluate the failure identification and correction of the proposed framework. Next, we evaluate efficiency by depicting performance with respect to the number of iterations. Finally, we evaluate how the performance of the

method varies when the size of the registered frames is different from those used in the training. We validate generalisation by always conducting experiments in a cross-database manner, that is, by training only on one dataset and testing on different ones.

3.7.1 Evaluation measures

We validate sequence registration performance by evaluating the ability of a method to reduce the overall registration error and its tendency to generate drift errors. To identify and compare drift errors, we also illustrate sequence registration performance by visualising the error variation over time.

We measure accuracy in terms of registration errors. We measure the *registration error* $e_{s,t}$ of the t^{th} frame of the s^{th} sequence by measuring the error in the estimation of the canonical points (see Section 3.6):

$$e_{s,t} = \frac{1}{2}\sum_{i=1}^{2}\sqrt{||\mathbf{x}_{i,t} - \mathbf{W}(\mathbf{x}'_{i,t};\hat{\mathbf{p}}^s_t)||}, \tag{3.29}$$

where $\hat{\mathbf{p}}^s_t$ is the estimated transformation, $\mathbf{x}_{i,t}$ is a canonical point and $\mathbf{x}'_{i,t}$ is the canonical point after perturbation by a rigid motion $\mathbf{p}^s_t$. The *average error* $\bar{e}_s$ over a sequence s is:

$$\bar{e}_s = \frac{1}{T-1}\sum_{t=2}^{T} e_{s,t}, \tag{3.30}$$

where T is the sequence length. (Note that the error is measured with respect to the initial frame.) The *overall average error* $\bar{e}$ for a dataset is:

$$\bar{e} = \frac{1}{N_S}\sum_{s=1}^{N_s} \bar{e}_s, \tag{3.31}$$

where N_S is the number of sequences in the dataset. The *average drift error* $\bar{e}_{drift}$ is defined as:

$$\bar{e}_{drift} = \frac{1}{N_S}\sum_{s=1}^{N_S} e_{s,T}. \tag{3.32}$$

Since drift error accumulates over time, the registration error between the first and last frames of the sequences serves as a useful metric to measure drift [180]. Finally, we use the *percentage of converged frames* measure, c, which is commonly used for registration algorithms [12, 233]:

$$c = 100 \times \frac{|\{e_{s,t} : e_{s,t} < 1, s \in \mathbb{N}_{[1,N_S]}, t \in \mathbb{N}_{[2,T]}\}|}{N_S(T-1)}, \tag{3.33}$$

where $|\cdot|$ denotes set cardinality. The measure c is a useful alternative to the overall average error when the average error is biased by a few frames with a high registration error.

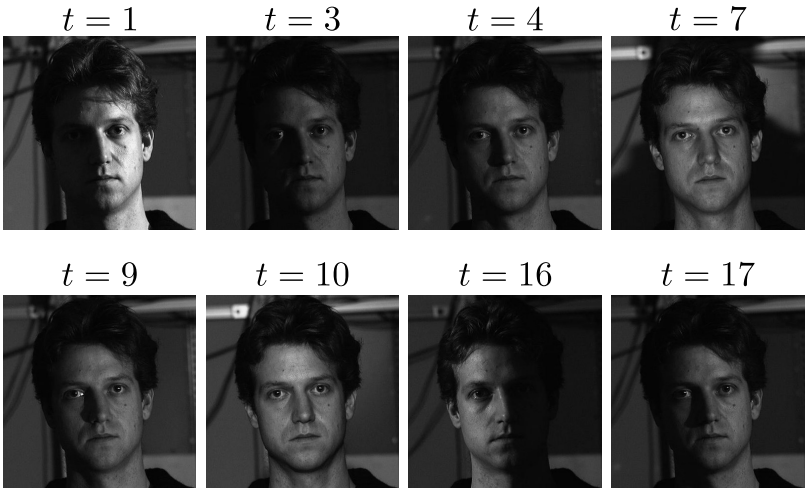


Figure 3.8: Sample frames from the PIE dataset. All the sequences in this dataset undergo similar illumination variations.

Following [12], we introduce a registration error to sequence frames by perturbing the canonical points with a random value drawn from a Gaussian white noise distribution with σ_{perturb} standard deviation. Since we focus on measuring registration accuracy and tendency to drift errors, we set $\sigma_{\text{perturb}} = 2$ when comparing with other methods, as LK methods may not converge for larger values [233]. However, we test our method with larger σ_{perturb} values when analysing the performance of our method for coarse-to-fine registration in Section 3.7.6.

3.7.2 Test datasets

To validate performance with real sequences of facial expression variations, we perform registration on three facial datasets: CK+ [131], MMI [168] and AFEW [47]. CK+ and MMI contains sequences of posed facial expressions of frontal faces. AFEW comprises sequences cropped from movies; the challenges of this dataset include out-of-plane head pose variations, illumination variations and background motion. Registered videos from these sequences are available for qualitative analysis on `ftp://spit.eecs.qmul.ac.uk/pub/es/s.zip`.

To quantify *robustness against illumination variations* we use the Pose, Illumination and Expression (PIE) dataset [209]. This dataset is collected from subjects that are sitting stably in front of a camera while the illumination conditions are changed rapidly in a controlled manner (see Fig. 3.8). We use 67 sequences (all the sequences that contain a frontal face). Each sequence is 21 frames long.

To quantify *robustness against non-rigid facial motions* we synthesised facial sequences with expression variations. We will refer to this as the *Synthesised* dataset. The need for such a

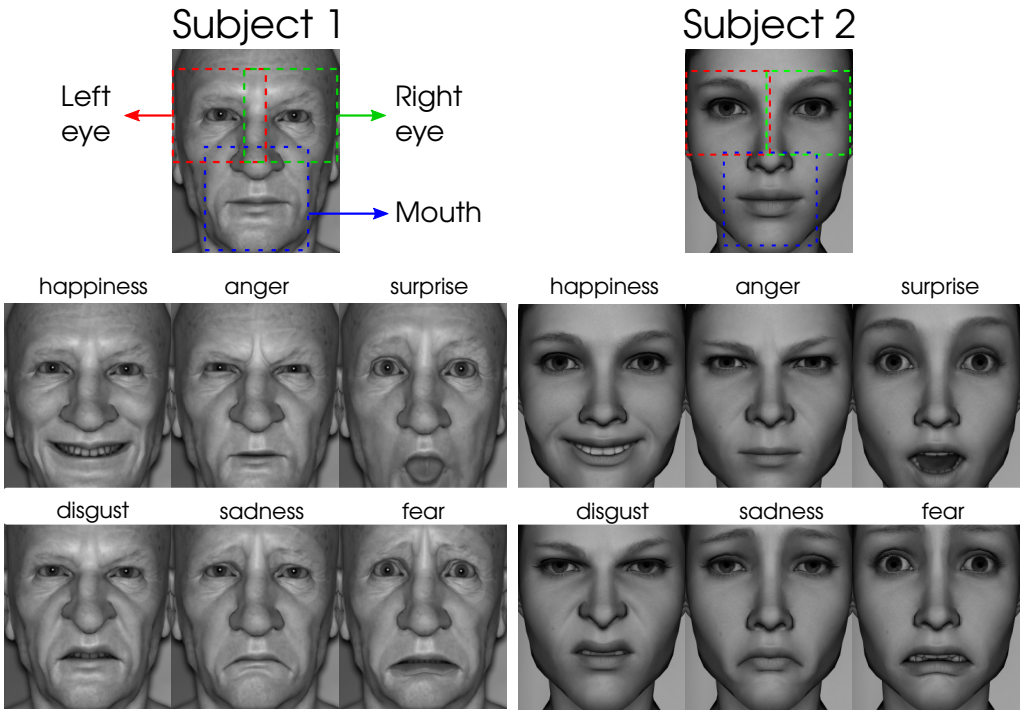


Figure 3.9: The apex frame of the six-basic expressions in the Synthesised dataset. The top-left facial image shows the cropping regions for part-based registration.

test sequence arises from the goal of having only expression variations without head or body movements. People tend to move while displaying an expression even in controlled datasets such as MMI [168] and therefore a ground truth for rigid registration cannot be obtained. To produce realistic faces, we use Autodesk Maya and two publicly available facial rigs[2], Old Man (Subject 1) and Ilana (Subject 2). Subject 1 is an old male with a wrinkled face, whereas Subject 2 is a young female who has a smooth skin (see Fig. 3.9). We created sequences that contain the six basic expressions by using the Action Units that are associated with those expressions. All sequences start with a neutral facial appearance, reach the apex, and then return to neutral appearance. We also include one sequence where there are no expression variations, thus yielding to a total of 14 sequences for the two subjects. The expressive-sequences of the Synthesised dataset are depicted in Fig. 3.10 and Fig. 3.11.

Prior to registration, we crop and resize the frames for all datasets. For whole-face registration, we first crop faces based on eye locations, and then resize the cropped frames to 200×200 pixels. For part-based registration, we first locate the centres of both eyes and mouth, and then crop each of these components so that the eye/mouth sits in the centre of frame after cropping. The cropped frames are then resized to 50×50 pixels. Fig. 3.9 illustrates the boundaries of

[2] `http://facewaretech.com/sdm_categories/rigs/`

Figure 3.10: Sequences of Subject 1; each column contains the sequence of one of the six-basic emotions. (a) Happiness, (b) anger, (c) surprise, (d) disgust, (e) sadness, (f) fear. To enhance visibility, we display only the part of the sequence between neutral and apex, and we skip every other frame.

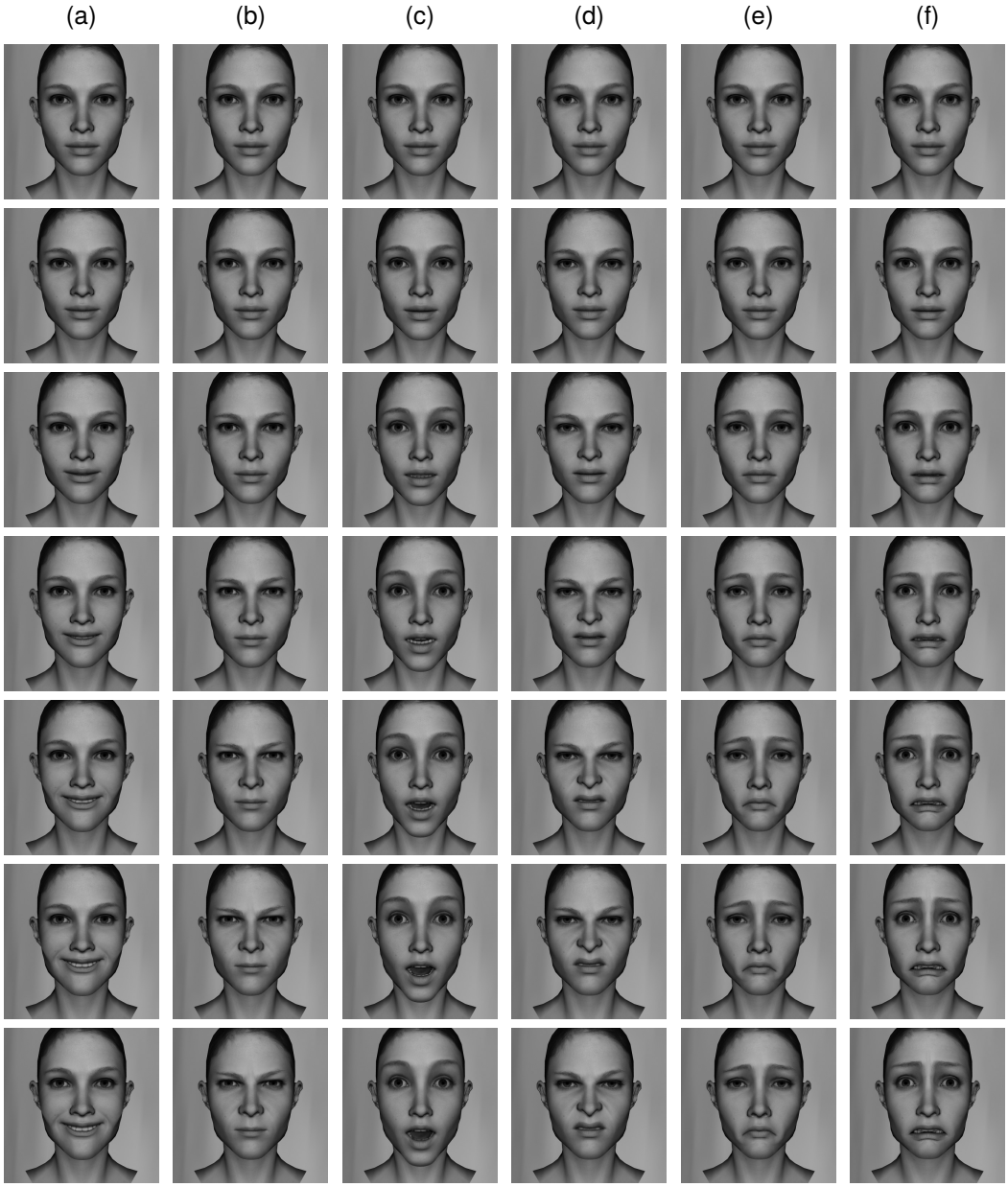


Figure 3.11: Sequences of Subject 2; each column contains the sequence of one of the six-basic emotions. (a) Happiness, (b) anger, (c) surprise, (d) disgust, (e) sadness, (f) fear. To enhance visibility, we display only the part of the sequence between neutral and apex, and we skip every other frame.

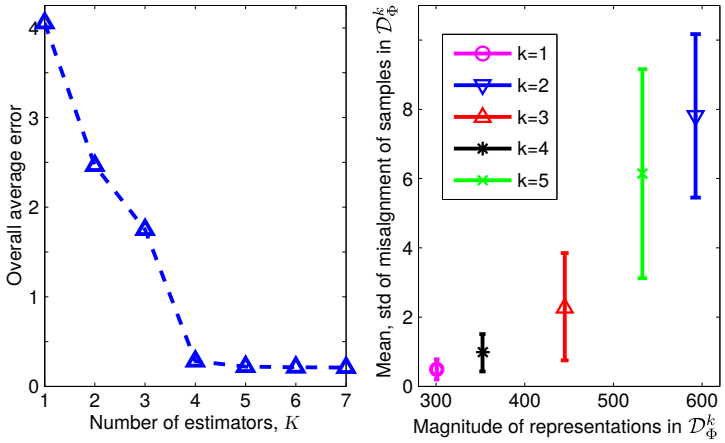


Figure 3.12: Left: average registration error against the number of estimators, K, suggests that $K < 4$ estimators are insufficient for accurate registration. Right: The mean and standard deviation of misalignment of samples in each dataset $\mathcal{D}_\Phi^k$ against the average magnitude of representations in $\mathcal{D}_\Phi^k$, highlights the coarse-to-fine structure of the set of 5 estimators.

the cropped components. For the Synthesised dataset we locate the centres of eyes and mouth manually, for the PIE and CK+ datasets we use the facial landmarks provided with the dataset. For the MMI and AFEW datasets we detect faces using OpenCV and locate landmarks using the Supervised Descent Method (SDM) [261].

3.7.3 Implementation details and parameter sensitivity

To compute the Gabor representation, we partition the energy matrices into $M \times M = 3 \times 3$ pooling subregions. We use standard deviation pooling (*i.e.* compute the standard deviation of the values in each subregion), as it outperforms mean and max pooling [189]. We use Gabor filters across 8 orientations, $\{0°, 45°, \ldots, 315°\}$ and 3 scales, $\{2^j\}_{j=0}^2$, yielding a filter bank with $K_G = 24$ filters, and an overall representation with $D = 9 \times 24 = 216$ features.

For optimisation we used the scaled conjugate gradients algorithm. We conducted the training on MATLAB® using the NETLAB [149] library and the testing on a C++ implementation. We set the convergence threshold ε_y to 1 pixel, which is the value used for the evaluation of the LK framework [12]. During correction, the width of the temporal window is set to $T_D = 5$ and we apply correction with a temporal window that considers also subsequent frames (*i.e.* online-with-delay). We created the training samples from CK+ [131] by perturbing frames from 20 sequences where we perceived no head or body motion. We fix the number of training samples to $N = 15,000$. We set the maximum number of iterations to $K_{\max} = 12$, which is sufficient for convergence for $\sigma_{\text{perturb}} = 2$ (see Section 3.7.1). Note that in Section 3.7.6 we analyse perfor-

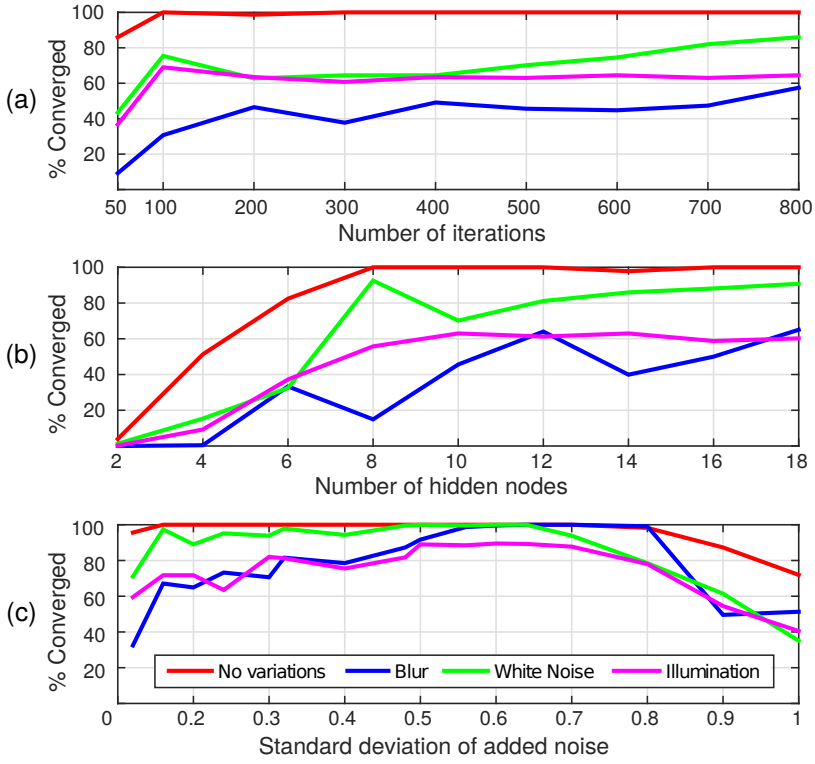


Figure 3.13: Registration performance against the (a) number of iterations, (b) number of hidden nodes and (c) σ_{noise} of the training samples. Adding noise to training samples with a σ_{noise} of approximately 0.6 enables the best generalisation against image blur, white noise and illumination variations.

mance against large registration errors with more iterations.

The number of estimators is $K = 5$, the number of hidden nodes in the neural network is $N_{\text{hidden}} = 10$, and the number of iterations is $N_{\text{iter}} = 500$. We will present below an experimental analysis that shows the effect of varying these parameters. For this purpose, we create testing data from the 6-basic expression sequences of the Synthesised dataset; specifically, we create misaligned image pairs (*i.e.* two-frame sequences) by picking all consecutive image pairs and perturbing the second image with $\sigma_{\text{perturb}} = 2$, and thus obtain $N_S = 228$ two-frame sequences.

Fig. 3.12 (left) reports the performance in terms of overall average error, $\bar{e}$, when varying K. The error is particularly high for $K = 1$, which suggests that a single neural-network cannot model the entire range of input-output mapping efficiently (see Section 3.5). In fact, when $K = 1$, the optimisation algorithm stops after only 26 iterations, which is a symptom of inefficient learning. Limited improvement is obtained when K is increased up to 3 as, similarly, training stops early. When $K = 4$ (and beyond) there is a significant performance improvement. In Fig. 3.12 (right) we illustrate the coarse-to-fine structure of the $K = 5$ estimators through the statistics of their corresponding datasets, $\{\mathcal{D}_{\Phi}^{k}\}_{k=1}^{5}$. Specifically, we show the average and the standard de-

viation of the registration error of all the samples in a $\mathcal{D}_\Phi^k$ against the average magnitude of the representations in $\mathcal{D}_\Phi^k$. Some estimators have a coarse structure as their training samples have large misalignment (*e.g.* $k = 2,5$), and others have a fine structure as their training samples have smaller misalignment (*e.g.* $k = 1,4$).

To prevent overfitting, it is useful to add a random noise to the motion representations computed from the training images. This noise is drawn from an isotropic zero-mean Gaussian distribution with standard deviation $\sigma_{\text{noise}} = 0.5$. Other well-known strategies to prevent overfitting are reducing N_{hidden} (*i.e.* using a simpler model) or reducing N_{iter} (*i.e.* performing early termination) [19]. Fig. 3.13 compares the efficiency of these three approaches in preventing overfitting, by providing the average convergence rate in the presence of three additional image variations. The first two variations are image blur with a Gaussian kernel of standard deviation 2, and additive white noise with standard deviation of 8 (similarly to [12]); these variations are applied to the second images of the test pairs created from the Synthesised dataset. The third variation is illumination, for which we created $N_S = 400$ two-frame test sequences from the PIE dataset. Fig. 3.13a shows that adjusting the N_{iter} parameter provides no performance improvement against image variations. The N_{hidden} parameter can be adjusted to improve performance against white noise, however, only limited improvement can be achieved against blur and illumination variations. Adding noise to training samples, on the other hand, can improve performance significantly; with $\sigma_{\text{noise}} = 0.5$, $\sigma_{\text{noise}} = 0.6$, the performance in the presence of blur, white noise or illumination variations becomes similar to the performance without those variations.

3.7.4 Methods under comparison

We compare the proposed framework, MUMIE, with a method from each of the categories listed in Table 2.2. We selected the robust methods with available software. In categories where there are multiple methods, we select experimentally the best-performing ones based on the preliminary experiments that we conducted; we considered only the best methods for the sake of compactness, as we report the detailed performance of the method by showing their error frame-by-frame for each sequence (see Fig. 3.18, 3.20, 3.21 and 3.22). Based on the afore-mentioned criteria, we finally compare with the following three methods. (i) The SURF-based method (Speeded Up Robust Features) as the keypoint-based method, which generally outperformed the MSER method (Maximally stable extremal regions). (ii) The Robust FFT (R-FFT) method [230] as the transformation-based method, which, to the best of our knowledge, is the only method that

Figure 3.14: Registration results for R-FFT, GradCorr our method, MUMIE, on a sequence with a disgust expression followed by blinking. MUMIE accumulates little drift error and is not affected by the sudden motions that occur during blinking.

proved robust against illumination variations and other outliers. (iii) The GradCorr method [233], which systematically outperformed three other LK methods in our pre-liminary we experiments (both for whole-face and for part-based registration), namely, IC-LK [12], ECC-LK [60] and Fourier-LK [10].

We also compare with SDM [261], a registration based on landmark localisation. Specifically, we perform registration by computing an Euclidean transformation based on the eye corners, which are useful reference points for rigid registration. However, SDM requires the entire face for localising landmarks and therefore we perform only whole-face registration.

3.7.5 Results and discussion

The results produced by MUMIE to register real sequences from the CK+, MMI and AFEW datasets with various types of facial activity (*e.g.* talking and facial expressions other than those of the six-basic emotions), out-of-plane head pose variations, occlusions and background motion are provided as supplementary material.

Fig. 3.14 shows registered frames from an 80-frame long MMI sequence that contains a disgust expression. The sequence contains also a blinking expression, which is a challenging quick facial action that may cause other algorithms to fail. MUMIE achieves accurate registration and a considerably smaller drift error.

Fig. 3.15 shows results from an anger sequence that contains a pitch rotation. Whole-face registration causes a downward motion around the eyes, which may be detrimental to the analysis of facial activity. When the eyes are registered independently, the effect of head rotation is reduced to a better extent. Sequences with head pose variation highlight the importance of doing part-based registration instead of whole-face registration.

We now quantify the benefits of using multiple reference frames and then compare MUMIE

Figure 3.15: Illustration that depicts the advantage of part-based registration for addressing out-of-plane rotations. The subject displays a small pitch rotation between the neutral phase ($t = 1$) and the apex phases ($t = 37$) of the expression. With whole-face registration (left), the effect of head-pose rotation is more evident, as the eye corners move visibly downwards in $t = 37$. The effect is less visible in part-based registration for left and right eye, as the eye corners are better aligned.

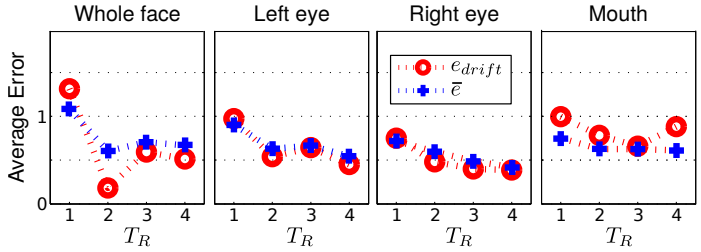


Figure 3.16: Average drift error, e_{drift}, and overall average error, $\bar{e}$, on the Synthesised dataset for varying numbers of reference frames, T_R.

with other methods.

Fig. 3.16 depicts the overall average error and average drift error on the Synthesised dataset when varying the number of reference frames, T_R. When $T_R = 2$ instead of $T_R = 1$ the error decreases consistently. The average registration errors for the whole-face, left eye, right eye and mouth decrease respectively to 13%, 55%, 64%, 78% when T_R is set to 2 instead of 1. When T_R is larger than 2 the error decreases generally at a lower rate and sometimes increases. The fact that performance saturates with $T_R = 2$ is desirable from a computational complexity perspective, as computation time increases with T_R. In the remaining experiments, we therefore set $T_R = 2$ while obtaining the multi-frame registration results for our method. The averaging that takes place when we integrate information from multiple frames as in (3.19) may be responsible in providing little improvement when $T_R > 2$. While taking the average has the advantage of keeping the input representation at a limited length that is independent of T_R, it also reduces the weight of each individual frame as T_R increases, since the average is computed by dividing the contribution of each frame with T_R.

Fig. 3.17 compares the *average registration error* of MUMIE with other methods on the Synthesised dataset. Overall, Fig. 3.17 suggests that MUMIE outperforms other methods sig-

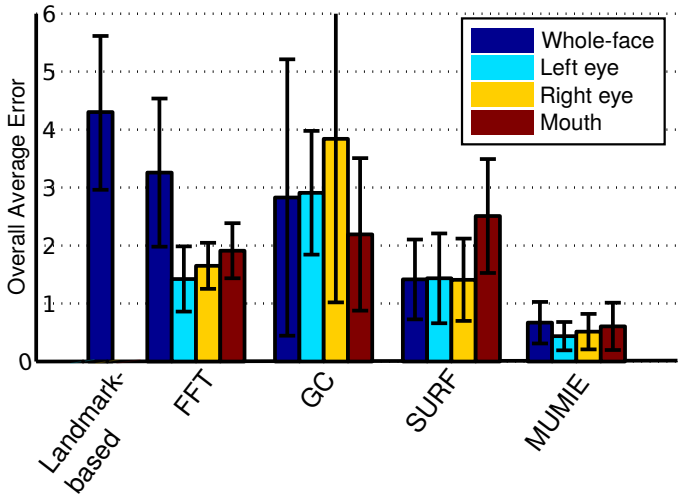


Figure 3.17: Sequence registration performance in terms of average registration error over 14 sequences (Synthesised dataset).

nificantly on sequences with facial expression variations. The error variation for each sequence over time is plotted in Fig. 3.18 for whole-face registration. Small landmark localisation errors among consecutive frames cause jittering when using landmark-based registration. Fig. 3.19 shows the difference between a sample pair of consecutive frames from the neutral sequence of Subject 1. A similar jittering can also be observed when using the R-FFT method. On the other hand, registration using SURF, GradCorr and MUMIE produces only little jittering, also for non-neutral sequences (registered sequences are provided as supplementary material). Even though the registration error may increase with expression variations (see Fig. 3.18), this increase happens gradually without a jittering effect, and the registration error at the end of the sequences becomes low, which is indicative of low drift error. The best results are obtained with MUMIE (multi-frame), and, expectedly, MUMIE (single-frame) produces higher drift errors compared to its multi-frame variant.

However, the whole-face registration performance of SURF and GradCorr do not generalise to part-based registration, as can be seen for the left-eye, right-eye or mouth sequences in Fig. 3.20, Fig. 3.21, Fig. 3.22, respectively. SURF keypoints are extracted from regions with rich texture. In part-based registration, frames contain less texture and relatively higher non-rigid motions; therefore, finding keypoints for rigid registration becomes more challenging. The part-based registration error of GradCorr generally increases gradually over time, however, unlike whole-face registration, the error is not reversed in the offset of the expression; therefore, part-based registration with GradCorr yields visible drift errors. While the failure of a generic

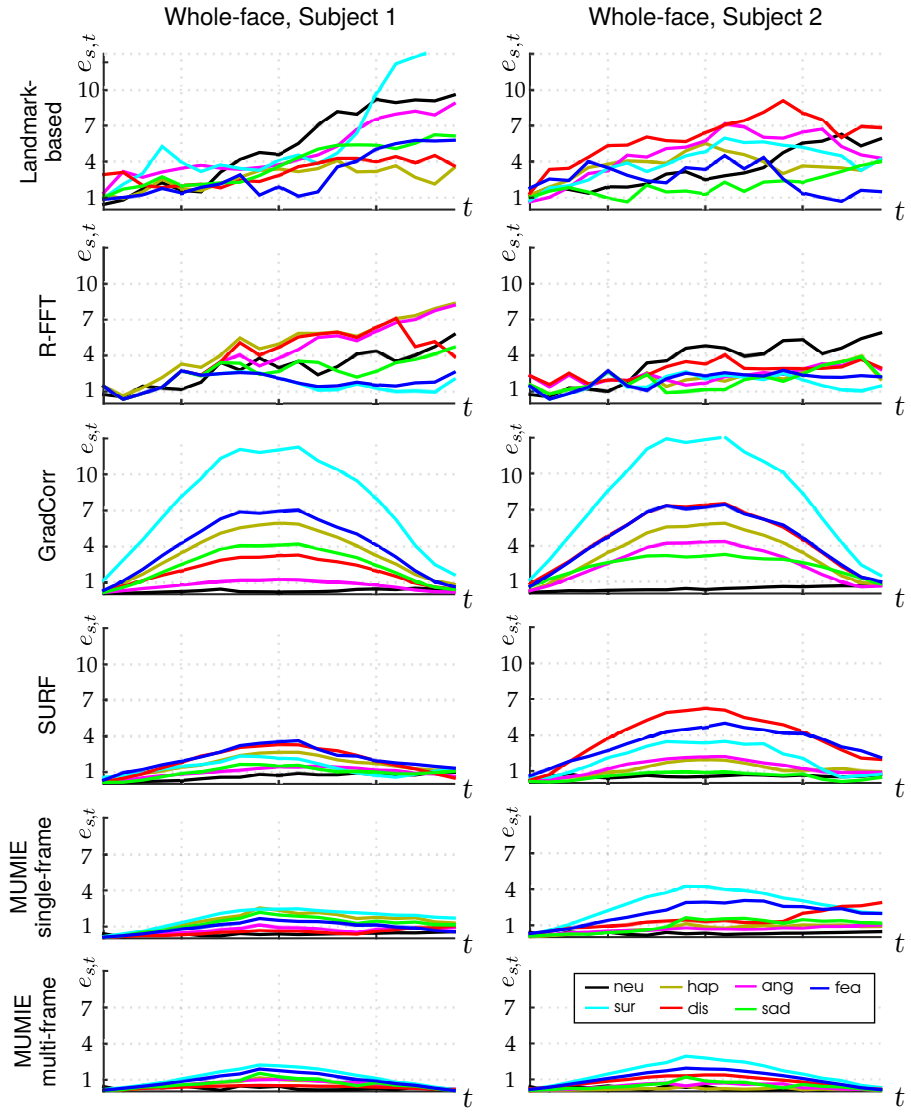


Figure 3.18: Registration error for whole-face sequences on the Synthesised dataset, depicted separately for the two subjects of the dataset and separately for each method. Each line represents the error over time, $e_{s,t}$, for one sequence (see legend for the expression in each sequence). MUMIE (multi-frame) results are obtained with $T_R = 2$.

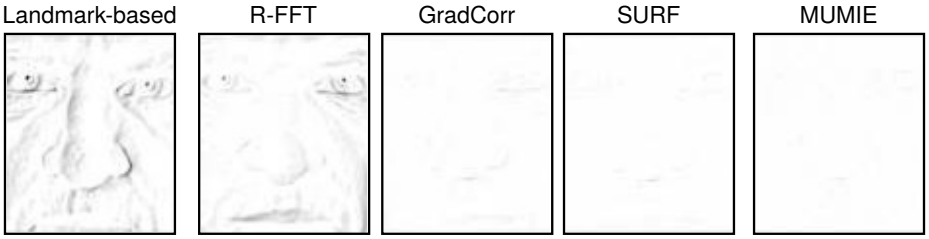


Figure 3.19: Difference images computed from a consecutive pair of images from the neutral sequence of Subject 1 of the Synthesised dataset. Grey levels visualise the registration errors. GradCorr, SURF and MUMIE produce little jittering error.

rigid registration method fwhen the input has considerable non-rigid variations is not surprising, the large error for the neutral sequences is an unexpected result. This may suggest that GradCorr

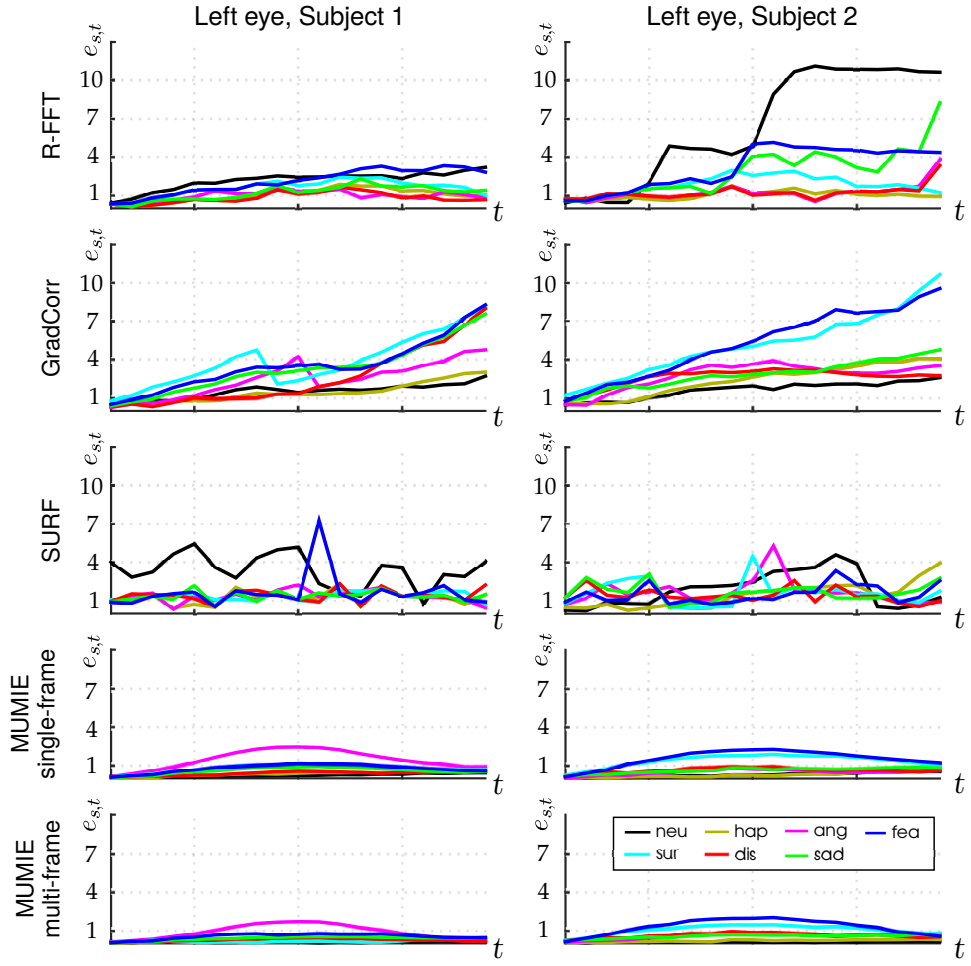


Figure 3.20: Registration error for left-eye sequences on the Synthesised dataset, depicted separately for the two subjects of the dataset and separately for each method. Each line represents the error over time, $e_{s,t}$, for one sequence (see legend for the expression in each sequence). MUMIE (multi-frame) results are obtained with $T_R = 2$.

requires some texture variation to operate reliably, even for simple cases without outlier motions.

Part-based registration is problematic also for R-FFT: see for example the large performance difference between the left and right eye of Subject 2 in Fig. 3.20 vs. Fig. 3.21. While investigating this irregular outcome further, we noticed a difference between the unregistered versions of the left and the right eye sequences. The initial registration error in left eye sequences of Subject 2 was causing the facial contour to appear in some frames and disappear in others. The R-FFT method operates on gradient images, and the contour of the face produces a high gradient which may be misleading the FFT-based algorithm.

The part-based registration errors of MUMIE are considerably smaller than those of other methods (Fig. 3.17 and Fig. 3.20, 3.21,3.22). The error for whole-face registration *does* generalise to part-based registration; that is, even though the error grows as the expression evolves

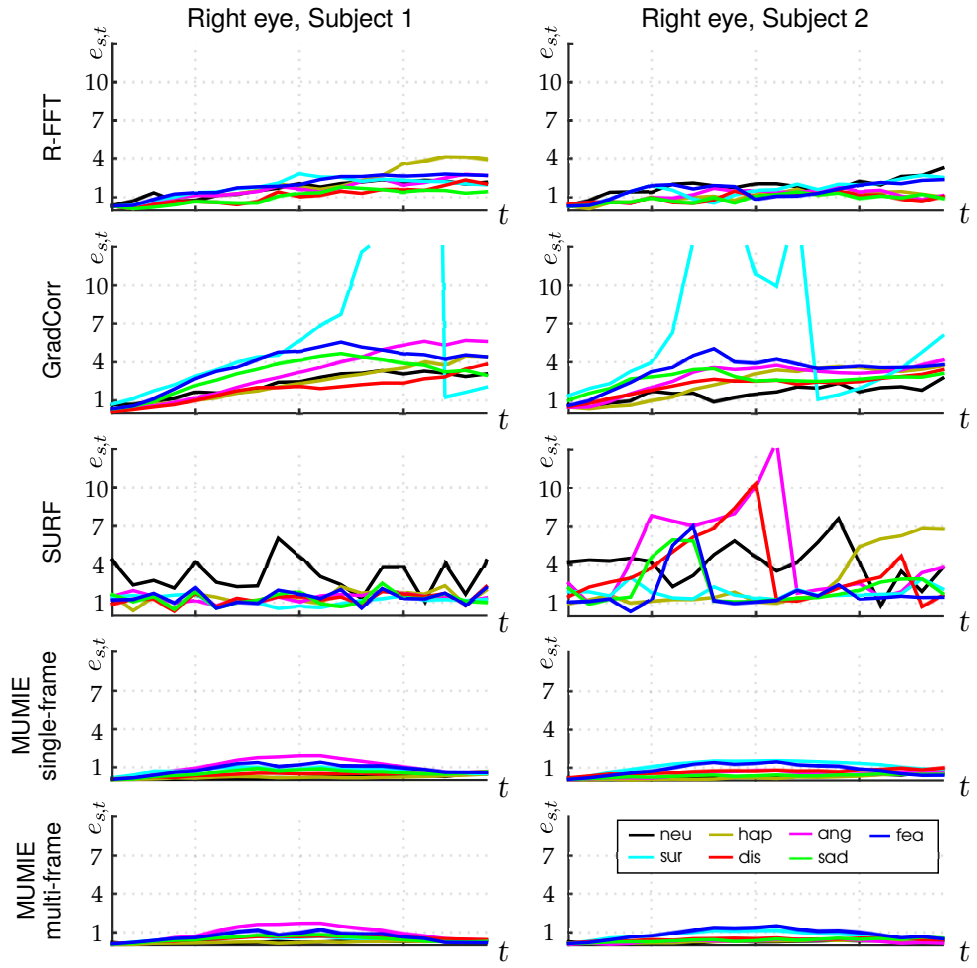


Figure 3.21: Registration error for right-eye sequences on the Synthesised dataset, depicted separately for the two subjects of the dataset and separately for each method. Each line represents the error over time, $e_{s,t}$, for one sequence (see legend for the expression in each sequence). MUMIE (multi-frame) results are obtained with $T_R = 2$.

to apex, the error decreases during offset. MUMIE outperforms other methods, even when it is used with a single reference frame, *i.e.* $T_R = 1$. The performance of MUMIE with $T_R = 2$ is particularly high, as the final error is less than 1 pixel for all sequences except the mouth sequences of Subject 1 and 2 for the surprise expression (see Fig. 3.22), which is the expression that involves arguably the largest non-rigid variation. The left and right eye performance of each subject is quite similar, which implies that symmetrical non-rigid motions of the same subject yield consistent results. The surprise and fear expressions cause higher errors when registering the eyes of Subject 2; this may be due to the eyebrows of Subject 2 being raised higher than those of Subject 1. (For both subjects, we raised eyebrows as much as possible when synthesising sequences, however, there are differences between facial rigs of the subjects.) Other inter-subject performance differences may be due to the skin texture; while Subject 1 has wrinkles, the skin of

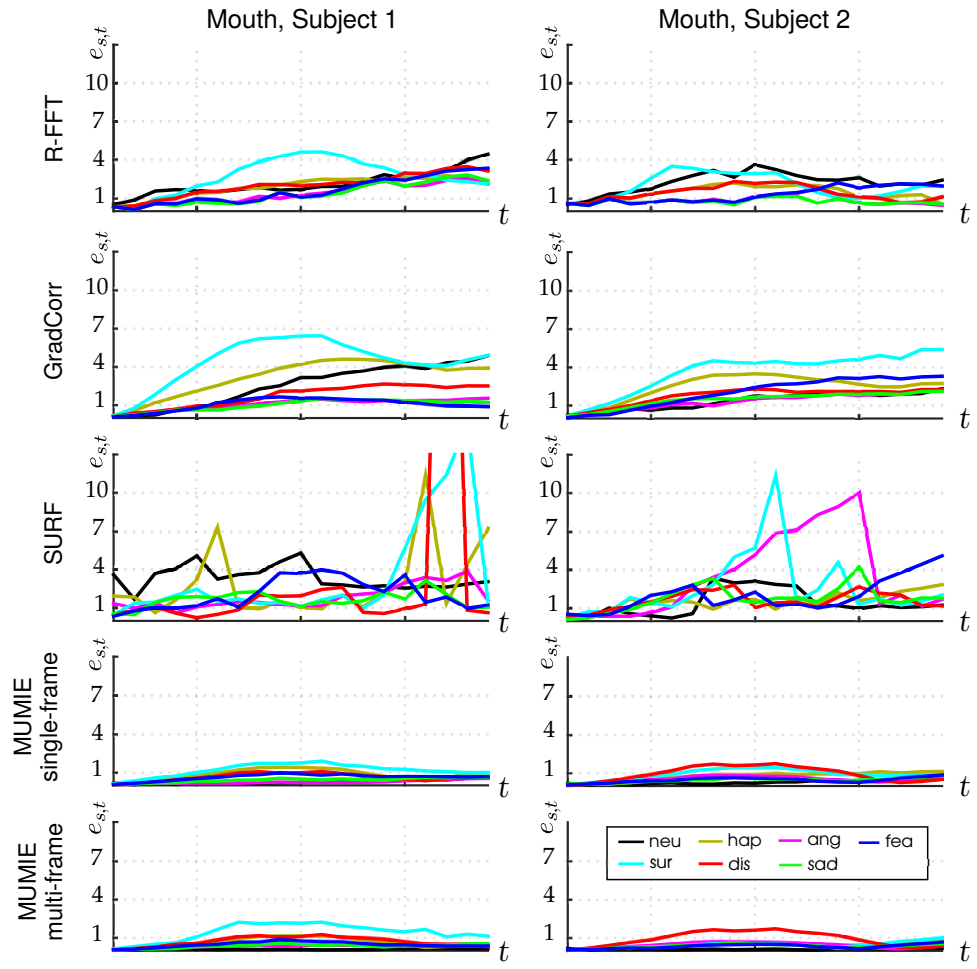


Figure 3.22: Registration error for mouth sequences on the Synthesised dataset, depicted separately for the two subjects of the dataset and separately for each method. Each line represents the error over time, $e_{s,t}$, for one sequence (see legend for the expression in each sequence). MUMIE (multi-frame) results are obtained with $T_R = 2$.

Subject 2 is smooth. Wrinkles may be advantageous as additional texture if they are not moved by the expression, or, they may be disadvantageous if they cause more non-rigid motion.

Non-uniform illumination variations typically cause registration failures on the PIE dataset, rendering the average sequence registration error of little use for quantitative evaluation. Therefore, we discuss the compared methods using the error visualised over time, where the performance of methods before and after failure is observed directly. Fig. 3.23 illustrates the performance of all compared methods for 5 randomly selected PIE sequences. (The results of all 67 sequences are shown as videos in the supplementary material.) Landmark-based registration deteriorates in the presence of illumination variations due to increased error in landmark localisation. SURF-based registration does not perform reliably in the PIE dataset as the number of matched keypoints falls significantly.

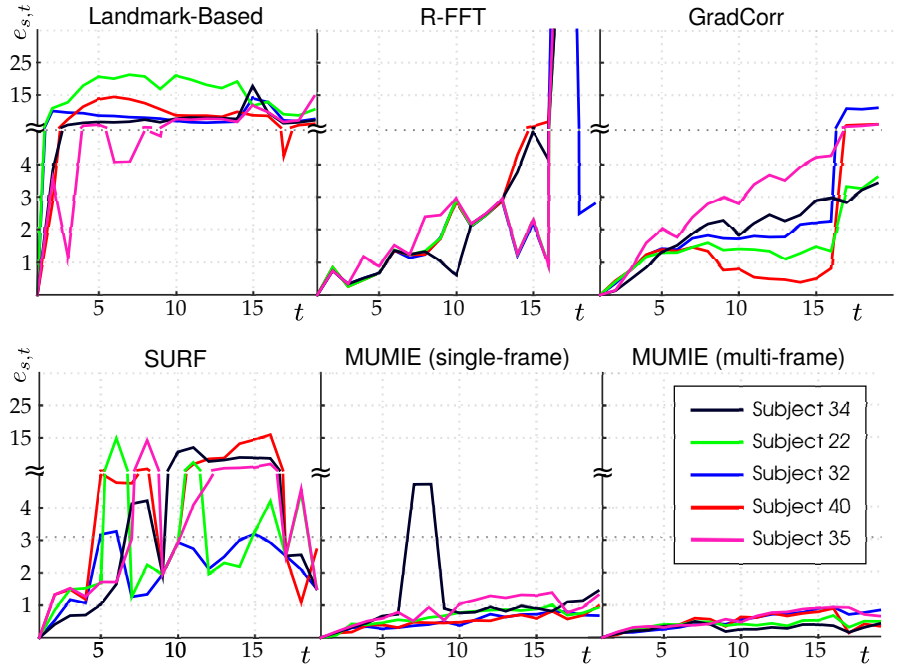


Figure 3.23: The performance of compared methods on five randomly selected PIE sequences, illustrated as error per frame over time, $e_{s,t}$. Each sequence is represented with a different colour. Note that we depict error at two different scales by inserting a break into the vertical axis. Even the robust R-FFT and GradCorr methods accumulate significant drift errors over time, whereas MUMIE produces little drift error, particularly in a multi-frame setting (*i.e.* with $T_R = 2$).

R-FFT is only slightly affected by illumination variations due to the robustness in the design of this method. R-FFT fails while registering the 17^{th} frame of almost all PIE sequences, due to the sudden illumination variation in this frame (see Fig. 3.8). GradCorr is also designed to be robust, and its performance deteriorates only slightly with illumination variations. Similarly to R-FFT, registration via GradCorr typically fails in the 17^{th} frames of the sequences. Compared to SURF or landmark-based registration, both R-FFT and GradCorr achieve significantly better performance in the presence of illumination variations. However, both methods accumulate drift errors over time.

Fig. 3.23 depicts the performance of MUMIE in after *failure identification and correction*. Uncorrected failures occurred only in two frames of Subject-34's sequence with MUMIE (single-frame). Overall, Fig. 3.23 suggests a considerable difference between MUMIE and other methods: the error is lower than that of other methods and, even though error does increase over time, the increase is lower than R-FFT or GradCorr. MUMIE (multi-frame) performs particularly well, as the error in the last frame is smaller than 1 pixel for all sequences.

Finally, Fig. 3.24 (bottom row) shows the average error $\bar{e}_s$ for each sequence of MUMIE (multi-frame) with and without failure identification and correction. The former is computed by

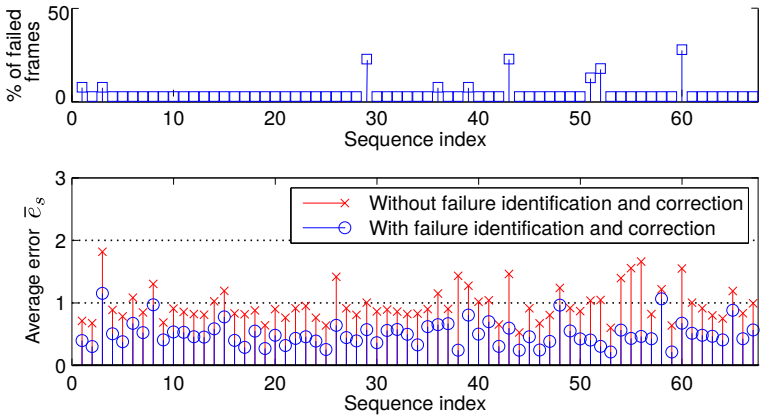


Figure 3.24: Performance of MUMIE (multi-frame) for each sequence of the PIE dataset. (Top): The percentage of successfully registered frames. (Bottom): The average registration error with and without failure identification and correction.

eliminating the frames where correction was not possible — Fig. 3.24 (top row) shows the ratio of those frames. The performance of our method is notably accurate after failures are automatically corrected, with an average error smaller than 1 pixel for 65 out of 67 sequences. Our method has successfully corrected most of failures of the PIE sequences (see Fig. 3.24 top). However, correcting a failure within a sequence may not be possible if a sudden appearance variation (*e.g.* out-of-plane head rotation) makes a frame visually dissimilar from all preceding frames. This may cause subsequent registration failures, and a reasonable action to take after a number of automatically-detected failures is to restart the registration process by changing the reference frame to the one where the sudden appearance variation caused the registration failure.

3.7.6 Computation time and convergence rate

We report the computation time of the proposed framework and highlight the usefulness of employing the magnitude of the motion representation as prior information while performing coarse-to-fine estimation.

Fig. 3.25 (left) shows the computation time per frame with respect to the amount of initial registration error. The overall average computation takes 2.74 seconds when all estimators are applied in a cascaded manner, and 1.59 seconds when estimators are selected at each iteration based on the magnitude of the motion representation (*i.e.* adaptively). For the cascaded approach, we allowed 6 iterations for the estimators except the finest one, as allowing for less iterations prevented convergence for some samples. The adaptive approach is on average faster, as coarse estimators are employed only when the initial registration error is large. Also, even when coarse

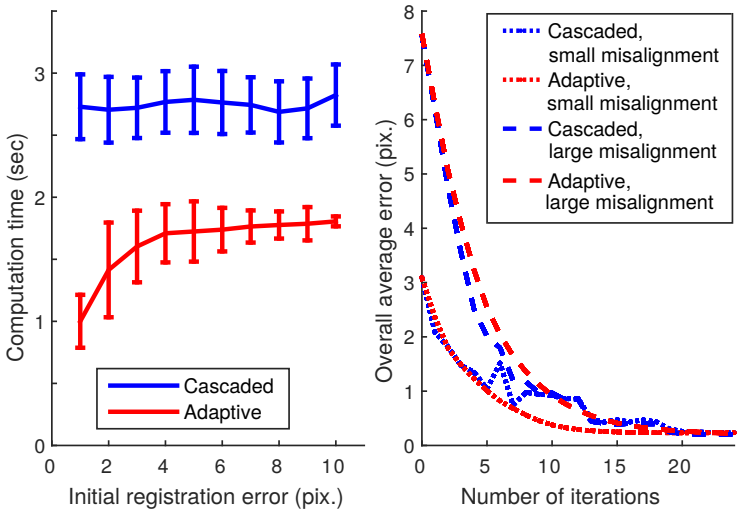


Figure 3.25: The efficiency improvement achieved by choosing the estimators based on the motion representation's magnitude (*i.e.* adaptively) instead of applying all estimators in a cascaded manner. Left: Computation time against initial registration error, shows that registration takes less time with the adaptive approach as coarse estimators are used only when misalignment is large. Right: Registration error against the number of iterations, depicted separately for samples of small misalignment and large misalignment. Note that error decreases monotonically with the adaptive approach.

estimators are used, they are generally used for less iterations, as we need not define a termination criterion for each estimator in the cascade.

Fig. 3.25 (right) highlights the advantage of the adaptive approach by showing the error against the number of iterations on two different test sets: one that includes samples with a registration error up to 4 pixels (*i.e.* small misalignment) and one where the registration error of samples reaches up to 15 pixels (*i.e.* large misalignment). As registration error decreases, the magnitude of the representation also decreases, and therefore the adaptive approach proceeds registration with finer estimators, which results in a monotonic decrease in average error, and an earlier convergence compared to the cascaded approach when misalignment is small. With the cascaded approach, error is not always reduced monotonically as in some cases the coarse estimators reach the limit of their granularity before they reach their limit of iterations, in which case they cannot reduce the registration error further. The cascaded and adaptive approaches reduce error at a similar rate on the set with samples of large misalignment. However, the cascaded approach is still slower on average, as in some cases the convergence occurs before the last (*i.e.* finest) estimator, yet, the cascaded approach needs to proceed with the subsequent estimators in the cascade at least for one iteration.

The maximal amount of registration error that can be tackled by our method depends on the

Table 3.1: Convergence rate against the amount of registration error. A representation computed from Gabor filters across 5 scales, $\{2^j\}_{j=0}^4$, can tackle larger registration errors than one computed from filters at 3 scales, $\{2^j\}_{j=0}^2$.

	Amount of registration error (pixels, ± 1)							
	4	6	8	10	12	14	16	18
Convergence rate with 3 scales	100	100	100	66.7	38.9	16.7	16.7	0.0
Convergence rate with 5 scales	100	100	100	100	100	100	100	79.2

scale of the Gabor filters that we use to compute the representation. We used filters at three scales, $\{2^j\}_{j=0}^2$, which may not converge if the registration error is 10 pixels or larger (see Table 3.1). However, larger filters can tackle larger errors, as we report in Table 3.1, where we trained a model that is based on a representation computed from filters at five scales, $\{2^j\}_{j=0}^4$.

3.7.7 Sensitivity to image size

During training we use frames with a specific size (*e.g.* 200×200 for whole-face, see Section 3.7.3). This section shows how the performance of the method varies when it is applied to frames with different sizes. The component of our framework that allows its usage with differently-sized images is the pooling of the representation (see Section 3.4.3), which makes the representation's dimensionality size independent. The type of pooling we employ is standard deviation (Section 3.7.3); below we first show how the output of standard deviation varies with image size, and then quantify registration performance.

As an example, let ϕ be the standard deviation of the top-left subregion of an energy matrix that is partitioned into 3×3 subregions. Fig. 3.26a visualises the ϕ computed from an image pair that has been rescaled to various sizes; specifically, the figure shows how ϕ varies against the amount of misalignment (*i.e.* vertical translation). ϕ is sensitive to image size because the thickness of the edges in an image changes as the image is resized and this affects the outcome of the convolution with the Gabor filters. However, the encoding of the misalignment is possible with all image sizes, because ϕ has a similar variation against the amount of translation for any size. Moreover, as the misalignment gets smaller, the difference between the ϕ values of differently sized images also gets smaller (and therefore an iterative registration scheme can converge). To illustrate this, Fig. 3.26b,c shows the registration performance of our method (trained only with 200×200-sized frames) on test sets with different image sizes. For these tests,

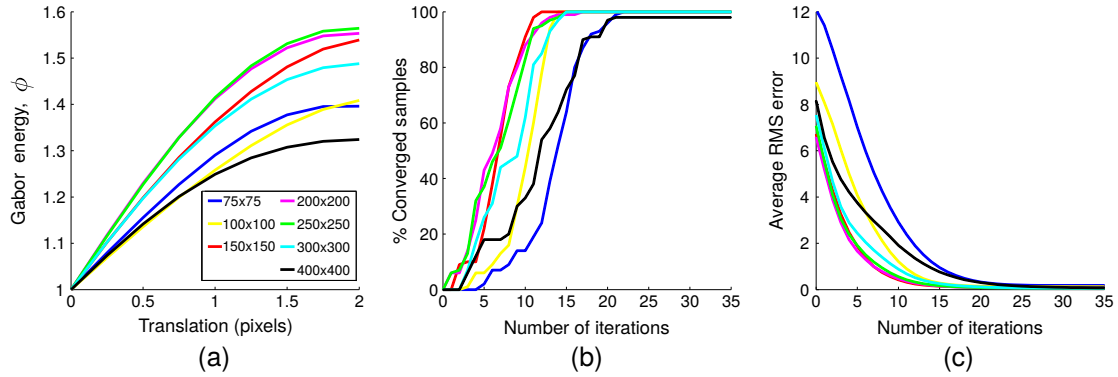


Figure 3.26: Impact of image size on the performance of the proposed method. (a) Variation of ϕ against the amount of misalignment, where ϕ is the standard deviation of the values within the top-left subregion of an energy matrix; the energy values are computed with a Gabor filter pair of scale 2 and orientation $\frac{\pi}{2}$. (b) Convergence ratio for each tested image-size (note that the proposed method was trained only with 200×200-sized images). (c) Average RMS error against the number of iterations for the same samples in (b); the error is computed only from samples that converged after 25 iterations.

we generate 100 test pairs for each tested image-size by perturbing the canonical points with a noise of $\sigma_{\text{perturb}} = 4$. Although registration needs more iterations for some image-sizes than for others, in most cases all test samples are registered with 100% convergence rate. The only exception is for 400×400 images whose convergence rate is 98%.

3.8 Limitations

The proposed method can tackle head pose variations only to a degree; although an initially non-frontal head pose does not cause problems even though the model is trained with frontal faces, head pose variation within the sequence can cause registration failures with sudden variations. Sudden head pose variations cause repetitive registration failures, and a useful strategy is to restart registration with a new reference frame after a number of consecutive failures.

While the Gabor motion energy is critical for accuracy and robustness against illumination variations, the computation of energy is involved and compromises real-time performance on conventional computers or mobile devices. Since convolution with Gabor filters is ubiquitous in computer vision, researchers considered efficient hardware-based solutions for this process [178]. Applying such solutions can be a straightforward approach to making the proposed registration framework computationally efficient. An alternative future direction is exploring and designing efficient approximations of speed- and orientation-selective Gabor filters [77].

3.9 Summary

We proposed a novel rigid registration framework based on optimisation via statistical learning that can cope with outlier non-rigid facial motions, drift errors and registration failures. Extensive experiments showed that using multiple reference frames during registration reduces drift errors and the proposed framework performs accurate registration in the presence of facial expressions or non-uniform illumination variations. Overall, the proposed framework performs reliably and consistently across various scenarios, both for whole-face or part-based registration.

The proposed registration framework will be used in the next chapter to register the training and test sequences for the unsupervised representation learning scheme.

Chapter 4

Bases of facial activity

4.1 Introduction

Having described sequence registration, we now proceed with the description of the proposed unsupervised representation learning framework. The proposed framework addresses three of the major issues that were identified for learnt spatio-temporal representations (see Section 2.5.6). That is, the framework is designed to recognise subtle expressions as well as pronounced expressions, to have little sensitivity to temporal inconsistencies (*i.e.* frame rate and observed temporal phases) between training and test sequences, and to be applicable with automatic recognition pipelines where the labels of the test sequences are not included in the sequences where the representation is learnt from.

Our framework is inspired from the standard facial representation that has been used in psychology even before computer-based analysis, namely, the FACS [57]. FACS is similar to a dictionary of elementary facial movements (*i.e.* AUs) that can be assembled into 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 a pronounced version of the same facial movement. Owing to those properties, FACS can describe nearly 7,000 expressions [197] as a combination of only 51 AUs [58].

Our representation is designed to mimic those two properties of FACS. Specifically, it is based on learning a linear model in which basis functions correspond to localised facial movements and basis coefficients relate to movement intensity. A representation learnt in this way has a number of generalisation advantages. Firstly, the fact that coefficients are designed to be proportional to intensity renders the learnt representation suitable for analysing expressions across a range of intensities, from subtle to pronounced expressions. The same property also allows for the usage of the representation when there are differences between the frame rates of the training and test videos (see Section 4.3). Furthermore, since the framework is unsupervised, the representation learnt using sequences having a specific set of expression labels (*e.g.* pronounced six basic expressions [167]) can be used with an automatic recognition pipeline to recognise expressions with other labels (*e.g.* three classes of micro-expressions [114]). We refer to the proposed representation as *Facial Bases*.

We first discuss our motivation for building a representation that mimics properties of AUs rather than recognising the AUs themselves (Section 4.2). Then we formulate the problem of learning Facial Bases (Section 4.3), describe our framework (Section 4.4) and the optimisation for its implementation (Section 4.5). Next, we visualise the bases by synthesising sequences (Section 4.6). We then elaborate on the main advantages of the representation by visualising the representation coefficients computed from real sequences (Section 4.7). We finally describe how Facial Bases can be used for automatic expression recognition (Section 4.9), the implementation details (Section 4.10) and the experimental results for automatic expression recognition (Section 4.11).

The technical contributions we present in this chapter are the following: we show that (i) to learn a linear model where basis coefficients are proportional to movement velocity, we must convert sequences into a representation where monotonic increases in movement velocity correspond to monotonic variations; and (ii) basis functions that correspond to localised facial activity can be learnt by training a sparsity-imposed linear model with Gabor phase shifts 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 draws upon the 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 ba-

sis functions are suitable for arguing for the existence of motion-sensitive but shape-insensitive representation in the human visual cortex [25]. On the contrary, our localised bases are shape-selective as each basis pertains to a specific facial region.

4.2 Comparison to automatic AU recognition

As we discussed in Chapter 2, the recognition of AUs is a popular research problem in automatic facial expression analysis. Naturally, one may consider that constructing a representation that mimics some properties of AUs may not be necessary if AUs themselves can be identified reliably. The output of multiple AU recognisers can be combined into an intermediate representation; this representation would be similar to what we aim to develop, and it could be used for higher-level recognition tasks, such as the recognition of the six basic emotions [245] or pain [129].

However, automatic AU recognition is a difficult problem on its own, particularly when it comes to recognising the intensities of the AUs. The main difficulties stem from the fact that AUs are defined in human (natural) language, and transferring their definition into a machine language is a non-trivial *supervised* machine learning problem. On the other hand, learning localised facial movement patterns from data in an *unsupervised* manner eliminates the need to transfer human knowledge to machines, as the bases are defined directly in machine language.

Some of the challenges related to constructing automatic AU recognisers can be listed as follows. 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 [37]. 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 [247]. 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 [35]. Discovering useful mappings via statistical learning from annotated data is another challenge, due to data imbalance between positive and negative samples [90] or to generalisation across subjects (*i.e.* identity bias) [190]. As a result, the recognition of AUs, and their intensities in particular, is still an open problem.

4.3 Problem formulation

Let $\mathbf{S} \in \mathbb{R}^{X \times Y \times T}$ 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 rigid registration technique (e.g. the technique discussed in Chapter 3). Moreover, let us assume that the motion between two consecutive frames of the sequence, I_{t-1} and I_t, is due to facial expression variations only[1]. Let $f(I_{t-1}, I_t)$ be a function that represents the motion between I_{t-1} and I_t at a local level (*e.g.* an optical flow function).

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:

$$f(I_{t-1}, I_t) = \sum_{k=1}^{K_A} A_k u_{t,k} + \varepsilon_t, \tag{4.1}$$

where ε_t represents reconstruction error. We want the basis coefficients, $u_{t,k}$, to be proportional to movement velocity, as the latter relates to expression intensity (see Section 4.4.1). 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 I_{t-1}, I_t contains a slow (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 $\{A_k\}_{k=1}^{K_A}$ 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. 4.1).

4.4 The learning framework

4.4.1 Dynamic bases

Facial expressions increase in their intensity gradually until they reach their apex [58]. In this process, the velocity of a facial component first increases from zero to a peak, then decreases back to zero [211]. The peak velocity is expected to become higher when the expression is of larger intensity (see also Section 4.7). Importantly, the changes in the velocity are *gradual* [211]; to capture this aspect via Eq. (4.1), the magnitudes $|u_{t,k}|$ should also vary gradually over time. The bases A_k are fixed, which implies that for Eq. (4.1) to hold we must use a motion representation

[1] Whereas in Chapter 3 we represented a registered frame with $\bar{I}_t$, in this chapter for clarity we represent a registered frame with I_t. Similarly, in this chapter $\mathbf{S}$ represents a registered sequence

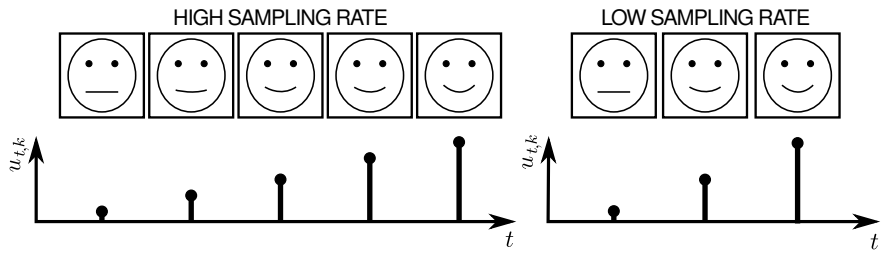


Figure 4.1: Illustration that depicts how a basis can provide useful information on videos with different frame rates. Let a basis A_k model the lip corner pulling that occurs during a smile. When a sequence is recorded at a lower rate, the apparent motion speed increases and the expression-related movement occurs at a higher apparent speed. If the basis coefficient $u_{t,k}$ is proportional to movement velocity as in Eq. (4.1), 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.

$f(I_{t-1}, I_t)$ whose elements are also changing gradually; that is, a monotonic increase in movement velocity should correspond to monotonic increases in $f(I_{t-1}, I_t)$ over time. Therefore, we cannot simply use the difference between the frames (*i.e.* derivative, $I_t - I_{t-1}$) 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 [257] or when computed from untextured regions such as cheeks [262].

To encode local motion without requiring the computation of motion vectors explicitly we chose to infer local motion through Gabor wavelets [62]. A frame I_t can be recovered from D_W complex Gabor wavelets $\{W_d\}_{d=1}^{D_W}$ as [113, 162]:

$$I_t = \sum_{d=1}^{D_W} \Re\{z_{t,d}^* W_d\}, \tag{4.2}$$

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

$$z_{t,d} = \rho_{t,d} e^{j\psi_{t,d}}. \tag{4.3}$$

Gabor wavelets have limited spatial support. The magnitude $\rho_{t,d}$ and phase $\psi_{t,d}$ take non-zero values when a visual element (*e.g.* an edge) within the wavelet's spatial support causes texture variation. The phase $\psi_{t,d}$ 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. 4.2). Since phase is sensitive to position, the *phase shift*

$$\dot{\psi}_{t,d} = \psi_{t,d} - \psi_{t-1,d} \tag{4.4}$$

is sensitive to motion [162]. Importantly, phase varies proportionally with the position, as shown in Fig. 4.2d. Since a Gabor wavelet W_d has local spatial support and is tuned to a specific orientation [113], the phase of one wavelet, $\psi_{t,d}$, 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, $\{W_d\}_{d=1}^{D_W}$, that span the whole image and are tuned to various orientations [113]. 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 [62].

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

$$\dot{\psi}_t = \sum_{k=1}^{K_A} A_k u_{t,k} + \varepsilon_t^u = \mathbf{A}\mathbf{u}_t + \varepsilon_t^u. \tag{4.5}$$

Note that this equation is a special form of (4.1). The term ε_t^u, which accounts for modelling errors, is assumed to be drawn from a von Mises distribution whose random variables, $\varepsilon_{t,d}^u$, are independent from one another and are modelled as $P(\varepsilon_{t,d}^u) \propto \exp[\kappa \cos(\varepsilon_{t,d}^u)]$ 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, $\mathcal{D}_{\dot{\psi}} = \{\dot{\psi}^n\}_{n=1}^N$, is the one that maximises the likelihood [163]:

$$\begin{aligned} P(\mathcal{D}_{\dot{\psi}}|\mathbf{A}) &= \prod_{n=1}^{N} P(\dot{\psi}^n|\mathbf{A}) \\ &= \prod_{n=1}^{N} \int P(\dot{\psi}^n|\mathbf{A},\mathbf{u}) P(\mathbf{u}) d\mathbf{u}. \end{aligned} \tag{4.6}$$

However, maximising $P(\mathcal{D}_{\dot{\psi}}|\mathbf{A})$ 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 [58]. 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 [163]. 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 $u_{t,k} - u_{t-1,k}$ [87]. Then the overall prior on $u_{t,k}$ becomes:

$$P(u_t|u_{t-1}) = \frac{1}{Z^u} e^{-\tilde{\lambda}_u \log(1+u_t^2)} e^{-\beta_u (u_t - u_{t-1})^2}, \tag{4.7}$$

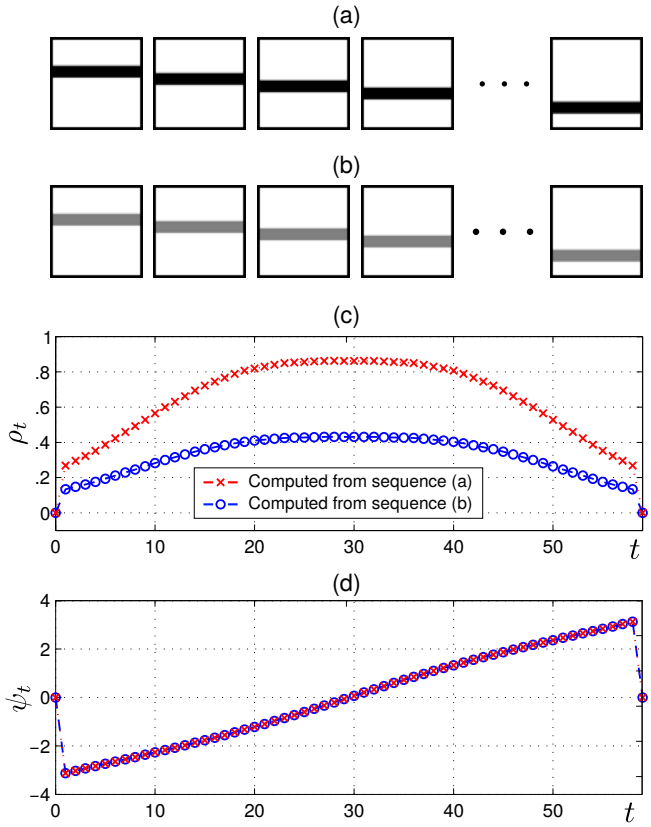


Figure 4.2: Illustration that highlights the ability of Gabor phase to encode motion. (a) Exemplar sequence that contains a horizontal bar moving vertically with a constant speed. (b) A sequence that is identical to the one in (a) except that the pixel intensity of the bar is multiplied by 0.5. (c) The 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, ψ_t, increases monotonically and is not sensitive to the intensity of the bar.

where $\tilde{\lambda}_u$ 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 [81] proposed a technique to produce such localised bases by enforcing the following sparseness metric:

$$\mathcal{S}(A_k) = \frac{\sqrt{D_W} - \frac{||A_k||_1}{||A_k||_2}}{\sqrt{D_W} - 1}, \tag{4.8}$$

where $||\cdot||_1$ and $||\cdot||_2$ denote the ℓ_1 and ℓ_2 norms, respectively. The sparser A_k, the higher the $\mathcal{S}(A_k)$. Sparse and localised bases are obtained by pre-defining a sparseness rate S_A and enforcing all bases to follow this rate (*i.e.* $\mathcal{S}(A_k) = S_A$) during optimisation (see Section 4.5).

4.4.2 Static bases

When there is no expression variation in a sequence, there is no motion and the phase shifts ψ_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. 4.3). This can be achieved by learning *static bases*, in a similar fashion to learning dynamic bases. Dynamic bases were learnt from phase shifts $\dot{\psi}_t$, whereas static bases are learnt from magnitudes:

$$\rho_t = (\rho_{t,1}, \rho_{t,2}, \ldots, \rho_{t,D_W}), \tag{4.9}$$

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, ρ_t, generated by any facial configuration. Specifically, we use the log-magnitudes, as taking logarithm linearises the dependencies between magnitudes [25]:

$$\log \rho_t = \sum_{k=1}^{K_B} B_k v_{t,k} + \varepsilon_t^v = \mathbf{B}\mathbf{v}_t + \varepsilon_t^v, \tag{4.10}$$

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

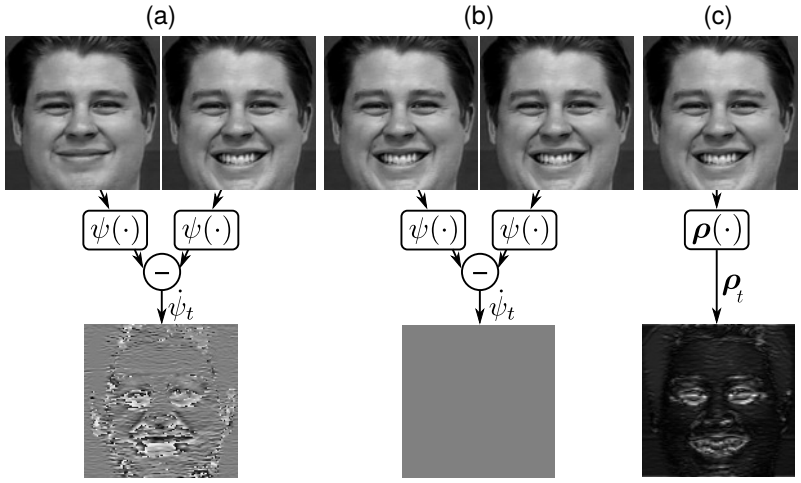


Figure 4.3: Example of the importance of the magnitude to recognize an expression. For clarity, magnitude and phase responses are illustrated only for one Gabor filter. (a) The phase shift provides useful information when there exist expression variations between consecutive frames. (b) The phase shifts 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.

The resulting prior $P(v_{t,k}|v_{t-1,k})$ is identical to Eq. (4.7) 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 $\tilde{\lambda}_v, \beta_v$ and Z^v. The overall pipeline of the proposed model is illustrated in Fig. 4.4. The variables referred to in Fig. 4.4 are listed in Table 4.1 along with their dimensionality.

4.4.3 On alternative motion encoding schemes

Although we encode motion using Gabor phase shifts, any motion encoding scheme that linearises motion can be used as the representation in the left-hand-side of (4.1), as we discuss in Section 4.4.1. That is, the encoding scheme is required to be such that a gradual and monotonic increase (decrease) in the movement speed of a facial component must correspond to a gradual and monotonic increase (decrease) in the encoded motion. The Gabor phase shifts representation satisfies this requirement and we have used it simply to start with a simple solution that proved to be useful [25]. Yet by no means Gabor phase shifts are the only solution. In fact, any optical flow method can be used, as the motion vectors satisfy the afore-mentioned requirement of providing quantities proportional to the speed of facial motion. However, the absence of texture in a smooth facial skin can cause optical flow failures. There have recently been proposed robust optical flow methods [181, 253] that are appropriate to substitute the motion representation in (4.1), including

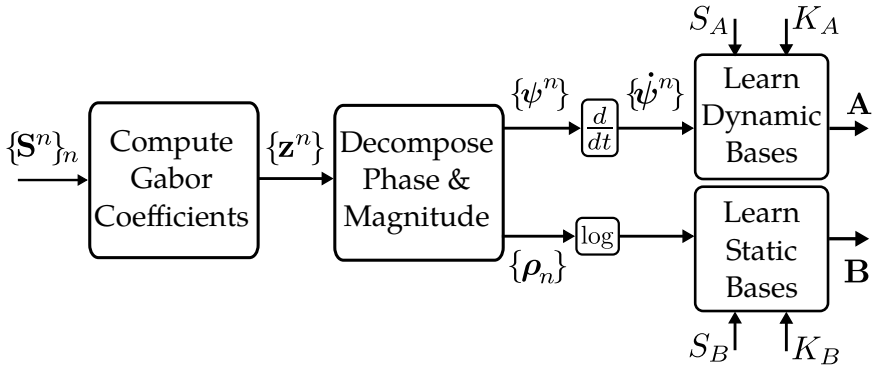


Figure 4.4: Illustration of how bases are learnt from a dataset, $\{\mathbf{S}^n\}_{n=1}^N$. The subscript n is dropped in later stages for clarity. The depicted variables are listed in Table 4.1 along with their dimensionality.

methods tailored for faces such as Face Flow [212]. Such methods could be used to improve the robustness of the proposed framework against illumination variations or occlusions.

4.5 Optimisation

4.5.1 Learning the bases

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

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

$$\mathbf{A}^* = \arg\max_{\mathbf{A}} \left[\log P(\mathcal{D}_\psi|\mathbf{A})\right], \tag{4.11}$$

$$\mathbf{B}^* = \arg\max_{\mathbf{B}} \left[\log P(\mathcal{D}_\rho|\mathbf{B})\right], \tag{4.12}$$

under the constraints

$$\mathcal{S}(A_k) = S_A,\ \forall k \in \{1,2,\ldots,K_A\}, \tag{4.13}$$

$$\mathcal{S}(B_k) = S_B,\ \forall k \in \{1,2,\ldots,K_B\}. \tag{4.14}$$

Maximising the likelihoods in Eq. (4.11–4.12) is equivalent to minimising the negative log-likelihoods, denoted as $\mathcal{E}_\psi = -\log P(\mathcal{D}_\psi|\mathbf{A})$ and $\mathcal{E}_\rho = -\log P(\mathcal{D}_\rho|\mathbf{B})$.

To minimise $\mathcal{E}_\psi$ and $\mathcal{E}_\rho$, we need their closed-form expressions, which are intractable due to the integrals such as those in (4.6). 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

Table 4.1: List of variables with their symbols and their dimensions.

$\mathbf{S} \in \mathbb{R}^{X\times Y\times T}$	An image sequence
$I_t \in \mathbb{R}^{X\times Y}$	t^{th} frame of $\mathbf{S}$
$\mathcal{D}_{\mathbf{S}} = \{\mathbf{S}^n\}_{n=1}^N$	Dataset to learn the bases from
$\mathbf{z} \in \mathbb{C}^{D_W\times T}$	Gabor coefficients of an $\mathbf{S}$
$\psi \in \mathbb{R}^{D_W\times(T-1)}$	Phase shifts computed from $\mathbf{z}$
$\rho \in \mathbb{R}^{D_W\times T}$	Magnitudes computed from $\mathbf{z}$
$\psi_t, \rho_t \in \mathbb{R}^{D_W}$	Phase shift, magnitude for t^{th} frame
$\mathbf{A} \in \mathbb{R}^{D_W\times K_A}$	Dynamic basis transformation matrix
$A_k \in \mathbb{R}^{D_W}$	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 $\mathbf{S}$
$\mathbf{u}_t \in \mathbb{R}^{K_A}$	Dynamic basis coefficients of I_t
$\mathbf{B} \in \mathbb{R}^{D_W\times K_B}$	Static basis transformation matrix
$B_k \in \mathbb{R}^{D_W}$	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 $\mathbf{S}$
$\mathbf{v}_t \in \mathbb{R}^{K_B}$	Static basis coefficients of I_t

integrals with the maximal value of their integrands [163]. Then, using the fact that $\varepsilon^u_{t,d}, \varepsilon^v_{t,d}$ are generated from von Mises and Normal distributions *respectively*, and using the priors $P(u_t|u_{t-1})$ and $P(v_t|v_{t-1})$, we can approximate $\mathcal{E}_\rho, \mathcal{E}_\psi$ as [25]:

$$\mathcal{E}_\psi \approx \sum_{n=1}^{N}\sum_{t=2}^{T}\sum_{d=1}^{D_W}\Big[\kappa\cos(\psi^n_{t,d} - [\mathbf{A}\mathbf{u}^n_t]_d) + \tilde{\lambda}_u \log(1+u^n_{t,d}) + \beta_u(u^n_{t,d} - u^n_{t-1,d})^2\Big], \tag{4.15}$$

$$\mathcal{E}_\rho \approx \sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{d=1}^{D_W}\Big[\frac{1}{\sigma^2_\rho}(\log\rho^n_{t,d} - [\mathbf{B}\mathbf{v}^n_t]_d)^2 + \tilde{\lambda}_v \log(1+v^n_{t,d}) + \beta_v(v^n_{t,d} - v^n_{t-1,d})^2\Big], \tag{4.16}$$

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. (4.6), we must follow a two-fold optimisation scheme [163]: 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. (4.15–4.16) 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):

$$\Delta A_{dk} = \kappa \sum_{n=1}^{N} \sum_{t=2}^{T} \sin(\dot{\psi}_{t,d}^n - [\mathbf{A}\mathbf{u}_t^n]_d) u_{t,k}^n, \tag{4.17}$$

$$\Delta B_{dk} = \sum_{n=1}^{N} \sum_{t=1}^{T} \frac{2}{\sigma_\rho^2} (\log \rho_{t,d}^n - [\mathbf{B}\mathbf{v}_t^n]_d) v_{t,k}^n. \tag{4.18}$$

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

$$\begin{aligned} \Delta u_{t,k}^n = & \kappa \sum_{d=1}^{D_W} (\sin \dot{\psi}_{t,d}^n - [\mathbf{A}\mathbf{u}_t^n]_d) A_{dk} \\ & - \tilde{\lambda}_u \frac{1}{2 + 2(u_{t,k}^n)^2} - 2\beta_u (u_{t,k}^n - u_{t-1,k}^n), \end{aligned} \tag{4.19}$$

$$\begin{aligned} \Delta v_{t,k}^n = & \frac{2}{\sigma_\rho^2} \sum_{d=1}^{D_W} (\log \rho_{t,d}^n - [\mathbf{B}\mathbf{v}_t^n]_d) B_{dk} \\ & - \tilde{\lambda}_v \frac{1}{2 + 2(v_{t,k}^n)^2} - 2\beta_v (v_{t,k}^n - v_{t-1,k}^n). \end{aligned} \tag{4.20}$$

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

$$u_{t,k}^n \leftarrow u_{t,k}^n + \alpha_u^{(\tau)} \Delta u_{t,k}^n, \tag{4.21}$$

$$A_{dk} \leftarrow A_{dk} + \alpha_A^{(\tau)} \Delta A_{dk}, \tag{4.22}$$

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 [149]. Efficient algorithms use learning rates defined automatically at each update step [149]. To this end, we use the Barzilai-Borwein method [17] 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. (4.13–4.14) can be satisfied with a number of ℓ_1 regularisation algorithms [264]. We use the projection algorithm proposed by Hoyer [81] as it has already proved successful in creating localised bases for facial data. We denote this projection algorithm as project$(\cdot)$ and use it to update A_k, B_k in order to satisfy Eq. (4.13–4.14) as:

$$A_k \leftarrow \text{project}(A_k; S_A), \tag{4.23}$$

$$B_k \leftarrow \text{project}(B_k; S_B). \tag{4.24}$$

Algorithm 2 Learn dynamic bases

Require: Dataset of facial videos $\mathcal{D}_{\mathbf{S}} = \{\mathbf{S}^n\}_{n=1}^{N}$, $\tau_A^{\max}$, $\tau_u^{\max}$

Ensure: Dynamic basis transformation $\mathbf{A} \in \mathbb{R}^{D_W \times K_A}$

1: Compute Gabor coefficients $\mathcal{D}_z = \{\mathbf{z}^n\}_n$ from $\mathcal{D}_{\mathbf{S}}$
2: Compute phases $\mathcal{D}_\psi = \{\psi^n\}_n$ from $\mathcal{D}_z$
3: Compute phase shifts from $\mathcal{D}_{\dot{\psi}} = \{\dot{\psi}^n\}_n$ from $\mathcal{D}_\psi$
4: Initialise A_k with random values $\forall k \in \{1, 2, \ldots, K_A\}$
5: Initialise $\mathbf{u}^n$ with random values $\forall n \in \{1, 2, \ldots, N\}$
6: $\tau_A \leftarrow 0$
7: **repeat**
8: **for** each sample $\dot{\psi}^n$ **do**
9: $\tau_u \leftarrow 0$
10: **repeat**
11: $\mathbf{u}^n \leftarrow \mathbf{u}^n + \alpha_u^{(\tau_u)} \Delta \mathbf{u}^n$
12: $\tau_u \leftarrow \tau_u + 1$
13: **until** $\tau_u^{\max}$ is reached
14: **end for**
15: **for** each A_k **do**
16: $A_k \leftarrow A_k + \alpha_A^{(\tau_A)}$
17: $A_k \leftarrow \text{project}(A_k; S_A)$
18: **end for**
19: $\tau_A \leftarrow \tau_A + 1$
20: **until** $\tau_A^{\max}$ is reached

Although Hoyer [81] originally proposed the algorithm for non-negative vectors, in the same paper he defines how the algorithm can be extended to be used for vectors that contain negative values too. The latter is critical for our work as the vectors that we aim to sparsify are the bases $\{A_k\}_{k=1}^{K_A}, \{B_k\}_{k=1}^{K_B}$ that can contain negative values. For the sake of being self-contained, we outline the project$(\cdot)$ algorithm in Algorithm 3. This is essentially identical to the second algorithm by Hoyer [81] except step 1 and step 20, which we included to explicitly show how the algorithm is applied for vectors with negative values. To summarise, the first step in Algorithm 3

is to store the signs of the values of the input vector in order to later apply them to the sparsified vector. The given vector is modified iteratively until its ℓ_1 and ℓ_2 norms reach the targeted values L_1 and L_2. The targeted ℓ_2 norm, L_2, is simply the initial ℓ_2 norm of the vector. The targeted ℓ_1 norm, L_1, depends on the targeted sparsification rate S and through (4.8) it can be computed as in step 3. The vector is updated through the projection that takes place in steps 8-10. If the updated vector contains negative values, then those values are set to zero and the process is re-initiated, otherwise, we have arrived at the desired solution. By the end of the process, the positive/negative signs that are stored in the beginning are applied to the sparsified vector. Note that in our application we aim to have basis vectors that are not only sparse (most of the elements to be zero) but also localised (most non-zero elements to be spatially connected). Although Algorithm 3 guarantees only sparseness, its application on facial images produced bases that are local too even though there is no explicit locality constraint [81]; therefore we have been motivated to primarily try this algorithm. The results that are shown below and in Section 4.6 show that this algorithm attains locality as well as sparsity also on the representation of Gabor phase shifts.

Fig. 4.5 exemplifies the learning process over iterations by visualising a training sequence and the evolution of the bases that take part in its reconstruction. Since the bases are initialised with random values (see Algorithm 2), they do not contain a particular structure after the first iteration, and the sequence that they reconstructed does not approximate the sequence reconstructed with the original phase values. In fact, all the frames of the sequence reconstructed with the estimated phases are nearly identical after the first iteration. Over the iterations, a structure emerges in the bases. After the 120^{th} iteration, the leftmost basis focuses on the movement around the inner part of the eyebrow, the basis in the middle focuses on the movement around the outer part of the eyebrow. The rightmost basis captures a movement located just below the other bases, which corresponds to the eyelid. The sequence reconstructed after the 120^{th} iteration shows an expression with a gradually increasing intensity, which is consistent with the intended purpose of the framework.

4.5.2 Inferring coefficients in a given sequence

Once the basis transformations $\mathbf{A}, \mathbf{B}$ are learnt, we compute the coefficients $\mathbf{u}, \mathbf{v}$ for a new sequence $\mathbf{S}$ as follows. First we compute the sequence's Gabor coefficients, $\mathbf{z}$. Then, we compute ψ and $\log\rho$ from $\mathbf{z}$. To obtain $\mathbf{u}$, we initialise $\mathbf{u}$ with random values as in step 5 in Algorithm 2,

Algorithm 3 The projection algorithm proposed by Hoyer [81]

Require: An input vector $\mathbf{x} = (x_1, x_2, \ldots, x_{D_x})$, a targeted sparseness rate S.

Ensure: The sparsified vector $\mathbf{s} = (s_1, s_2, \ldots, s_{D_x})$

1: Store signs $\kappa_x = (\kappa_1, \kappa_2, \ldots, \kappa_{x_{D_x}})$, i.e. $\kappa_i \leftarrow \text{sign}(x_i)$

2: Set $L_2 \leftarrow ||\mathbf{x}||_2$

3: Set $L_1 \leftarrow L_2 \times [\sqrt{D_x} - S(\sqrt{D_x} - 1)]$

4: Set $\mathbf{s} = (s_1, s_2, \ldots, x_{D_x})$ such that $s_i \leftarrow |x_i| + (L_1 - \sum_i |x_i|)/D_x$

5: Set $\mathcal{Z} \leftarrow \varnothing$

6: $k \leftarrow 0$

7: **repeat**

8: Set m_i such that if $i \in \mathcal{Z}$ then $m_i \leftarrow L_1/(D_x - |\mathcal{Z}|)$, otherwise $m_i \leftarrow 0$

9: Set $\mathbf{s} \leftarrow (s_1, s_2, \ldots, s_{D_x})$

10: Set $\mathbf{s} \leftarrow \mathbf{m} + \alpha(\mathbf{s} - \mathbf{m})$ where $\alpha \geq 0$ is set such that the resulting $\mathbf{s}$ satisfies the desired ℓ_2 norm value, L_2.

11: $\mathcal{Z}_k \leftarrow \{i : s_i < 0\}$

12: **if** $|\mathcal{Z}_k| = 0$ **then**

13: **break**

14: **end if**

15: Set $\mathcal{Z} \leftarrow \mathcal{Z} \cup \mathcal{Z}_k$

16: Set $s_i \leftarrow 0$, $\forall i \in \mathcal{Z}$

17: Set $s_i \leftarrow s_i - (\sum_i s_i - L_1)/(D_x - |\mathcal{Z}|)$, $\forall i \notin \mathcal{Z}$

18: Set $k \leftarrow k + 1$

19: **until** $k = D_x$

20: Set $s_i \leftarrow s_i \kappa_i$, $\forall i \in \{1, 2, \ldots, D_x\}$

and finally compute $\mathbf{u}$ iteratively as in steps 8–11 of Algorithm 2. A similar procedure follows for computing $\mathbf{v}$.

4.6 Synthesis for visualising the 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

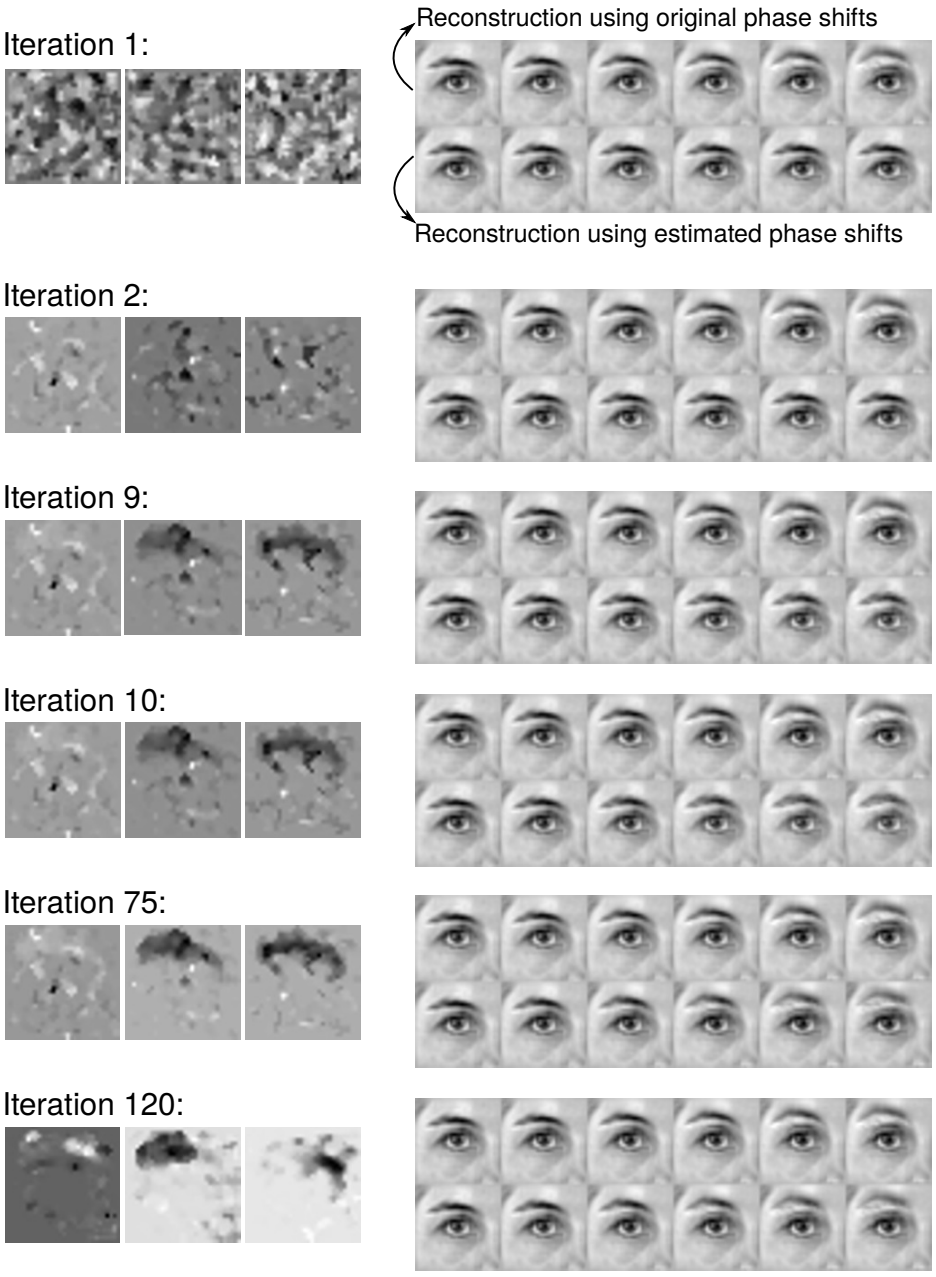


Figure 4.5: Illustration that depicts the learning of the bases over training iterations. On the left we illustrate three bases. On the right we depict a sequence that is reconstructed using (i) the original phase shift values and (ii) the phase shifts estimated through the proposed framework. To facilitate the visual interpretation we perform reconstruction using only the phase values of the Gabor coefficients, ignoring their magnitudes; and while visualising the bases, we consider only the basis values from Gabor wavelets at one scale and orientation and reshape those basis values into a square.

A_k, we first select a facial image, I_k^0, and then synthesise frames that reflect the movement encoded in A_k. Using Eq. (4.2–4.3), we can represent I_k^0 as:

$$I_k^0 = \sum_{d=1}^{D_W} \Re\{\rho_{k,d}^0 e^{-\psi_{k,d}^0} W_d\}. \tag{4.25}$$

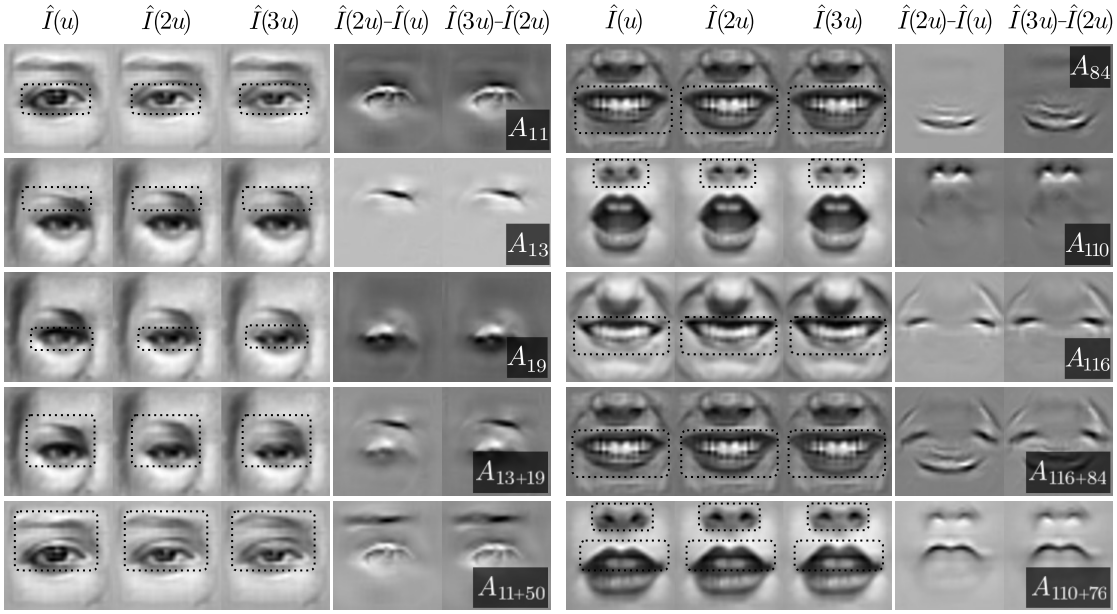


Figure 4.6: Illustration of the movement encoded in some of the dynamic bases. To illustrate a basis, A_k, or a combination of bases, A_{k+i}, we synthesise three images with three coefficients: $\hat{I}^0_k(u)$, $\hat{I}^0_k(2u)$ and $\hat{I}^0_k(3u)$. (Note that we drop the subscript k and superscript 0 for clarity.) We encircle the regions with facial movements, and provide the difference images of consecutive frames that also highlight those regions.

Synthesising an image amounts to altering the phase pattern, ψ^0_k, using phase shifts generated through Eq. (4.5) as:

$$\hat{\mathbf{I}}^0_k(u) = \sum_{d=1}^{D_W} \Re\{\rho^0_{k,d} e^{-j(\psi^0_{k,d} + [A_k u]_d)} W_d\}. \tag{4.26}$$

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

$$\hat{\mathbf{I}}^0_{k+i}(u) = \sum_{d=1}^{D_W} \Re\{\rho^0_{k,d} e^{-j(\psi^0_{k,d} + [A_k u]_d + [A_i u]_d)} W_d\}. \tag{4.27}$$

For representative purposes, we visualise bases learnt from the MMI dataset [167], which contains facial actions with their entire temporal evolution. We set the number of bases to $K_A = 60$ (see Section 4.11.3 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, I^0_k, from the CK+ dataset [96]. 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 [190] 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. 4.6 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.* $\hat{I}_k^0(2u) - \hat{I}_k^0(u)$ and $\hat{I}_k^0(3u) - \hat{I}_k^0(2u)$. 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. 4.6.

It is interesting to notice similarities between some AUs of FACS and the bases A_k shown in Fig. 4.6. 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[2].

Fig. 4.6 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. 4.7). 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.

4.7 Conceptual advantages of analysis via Facial Bases

In this section we highlight the two advantages of the bases. Firstly, the bases provide a plausible description of facial physiology that allows the usage of the same bases to identify pronounced

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

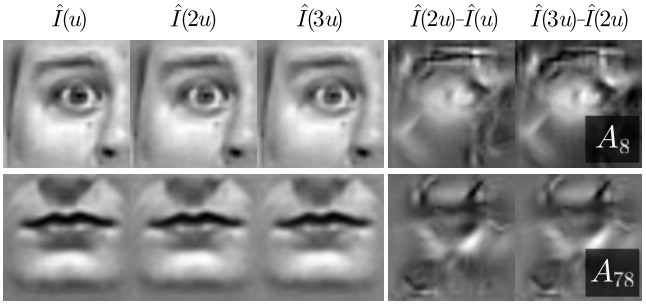
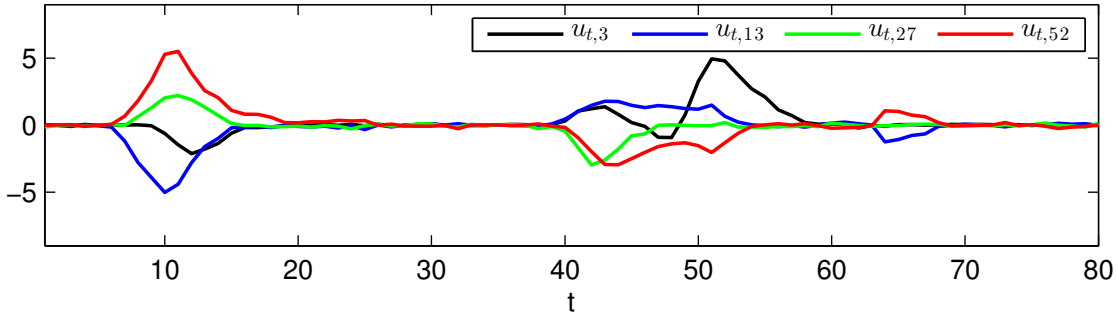


Figure 4.7: Sample bases that model non-localised texture variations.

(a) A left eye sequence, representing the emotion of disgust.

(b) Coefficients $u_{t,k}$ computed from the entire sequence above jointly.



(c) Coefficients $u_{t,k}$ computed for disjoint segments of the sequence in (a) independently.

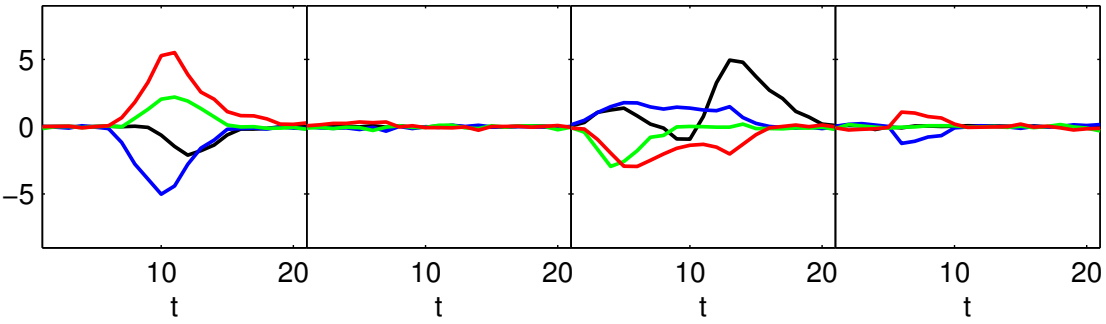


Figure 4.8: Illustration that depicts the coefficients $u_{t,k}$ computed from an exemplar sequence. Computing the sequence from the entire sequence or from its disjoint segments makes little difference as compared in (b) versus (c). For clarity, we depict only the coefficients $u_{t,k}$ obtained from the four most activated bases A_k. See Fig. 4.9 for the corresponding mouth sequence.

expressions *and* subtle expressions. While discussing this point, we will also comment on the identity bias of the bases. Secondly, the coefficients are consistent whether all the temporal phases of the expression are observed or not. As we will discuss, this is an important ability for representing naturalistic facial expressions.

Fig. 4.8b shows the coefficients computed from the MMI sequence displayed in Fig. 4.8a. The coefficients are near to zero whenever there is no facial activity; that is, in the neutral phase (*e.g.* between the frames $t = 1$ and $t = 5$) and also in the apex where the expression reaches a stable level (*e.g.* between the frames $t = 20$ and $t = 35$). (The corresponding mouth sequence and its coefficients are depicted in Fig. 4.9.) In moments of facial activity, the magnitude of coefficients undergo a smooth increase from (near) zero to a peak, and then they start decreasing also smoothly towards zero, consistently with the velocity of the movement produced by the muscles [108, 211]. This property of coefficients not only allows us to monitor the intensity variation *within* a sequence, but also to use the same bases to analyse *different* sequences across a variety of intensities, from micro-expressions to pronounced expressions. Fig. 4.10a depicts the movements from a micro-expression sequence of the SMIC dataset [114]. This sequence is labelled as 'positive', and contains a lip corner pulling that is indicative of happiness [57]. Our representation captures this movement with the activated $u_{t,116}$ coefficient. (Recall that A_{116} encodes a lip corner pulling as shown in Fig. 4.6). The sequence in Fig. 4.10b contains an even more subtle lip corner movement, hence the even smaller-magnitude (but non-zero) $u_{t,116}$ coefficients. On the other extreme, pronounced happiness expressions produce high-magnitude $u_{t,116}$ coefficients, as depicted in Fig. 4.11 and in Fig. 4.12. That the same basis, A_{116}, is activated for the same movement type across different datasets and subjects with different facial characteristics is a desirable outcome for defeating identity bias.

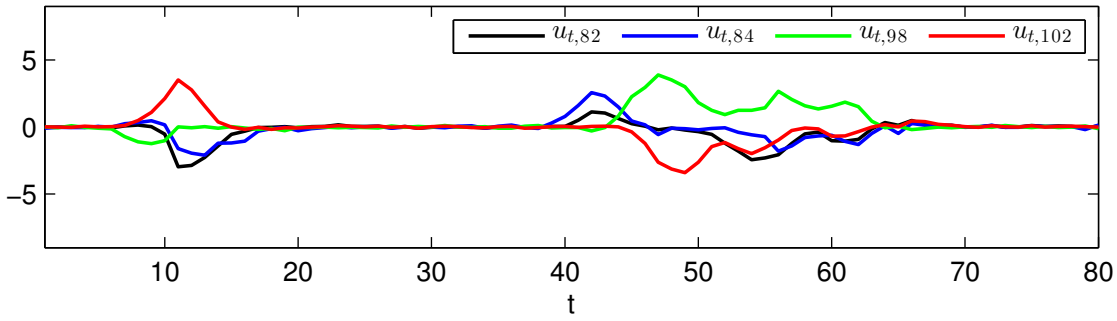
However, there is not always such a one-to-one mapping between facial movement types and bases. Fig. 4.13a and Fig. 4.13b show micro-expression sequences with a similar eyebrow movement. While the eyebrow movement is not very visible in Fig. 4.13a, we can identify it through the activated $u_{t,13}$ coefficients (see A_{13} in Fig. 4.6 and Section 4.6). However, the basis A_{13} is not activated in Fig. 4.13b, but the basis A_{25} *is* activated. In fact, A_{25} encodes a similar movement to the one in A_{13}, as can be seen in the supplementary videos[3]. Having multiple bases for one type of movement is probably due to identity bias; for example, the location, thickness or slanting of the eyebrows in different people may lead to having different bases. Fortunately, this has a limited impact on automatic facial expression recognition with the Facial Bases, because a unique encoding of different expressions — there is a one-to-many mapping.

Let us now proceed to the second advantage of the coefficients — that they do not require

[3]See `ftp://spit.eecs.qmul.ac.uk/pub/es/supp.zip`

(a) A mouth sequence, representing the emotion of disgust.

(b) Coefficients $u_{t,k}$ computed from the entire sequence above jointly.



(c) Coefficients $u_{t,k}$ computed for disjoint segments of the sequence in (a) independently.

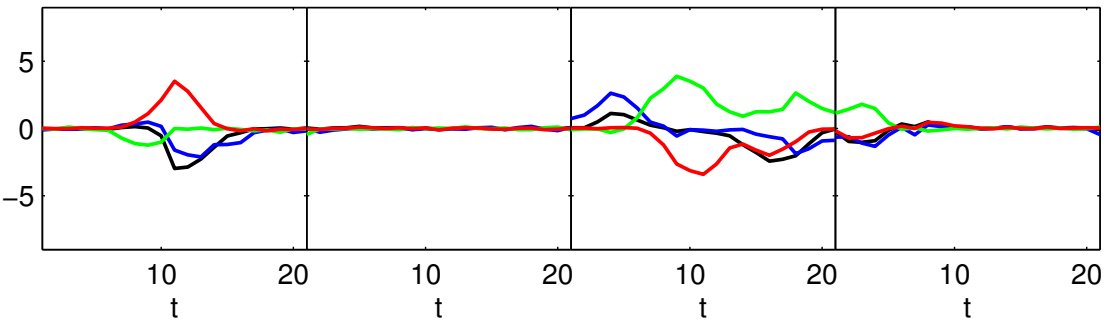


Figure 4.9: Illustration that depicts the coefficients $u_{t,k}$ computed from an exemplar sequence. Computing the sequence from the entire sequence or from its disjoint segments makes little difference as compared in (b) versus (c). For clarity, we depict only the coefficients $u_{t,k}$ obtained from the four most activated bases A_k.

the entire sequence for their computation. Since our model is linear, the sign of the coefficients $u_{t,k}$ controls the direction of the movement. The onset and offset phases of an expression can therefore be modelled with the same coefficients but reversed in their sign, as can be observed in Fig. 4.8b by comparing the onset (*i.e.* frames between $t = 5$ and $t = 20$) and the offset (*i.e.* frames between $t = 40$ and $t = 60$) coefficients. (Similar observations can be made for Fig. 4.9 and Fig. 4.11–4.12, and for the additional illustrations in Appendix B.1). This implies that the onset and offset are encoded uniquely. Importantly, we do not need to observe the neutral or onset phases of the expression to encode the offset in the same way. Fig. 4.8c shows the coefficients $u_{t,k}$ that are computed separately from four disjoint segments of the sequence in Fig. 4.8a. The $u_{t,k}$ in the third quadrant of Fig.4.8b are computed from a segment that starts with the apex, yet they are very similar to the corresponding $u_{t,k}$ that are computed from the entire sequence jointly

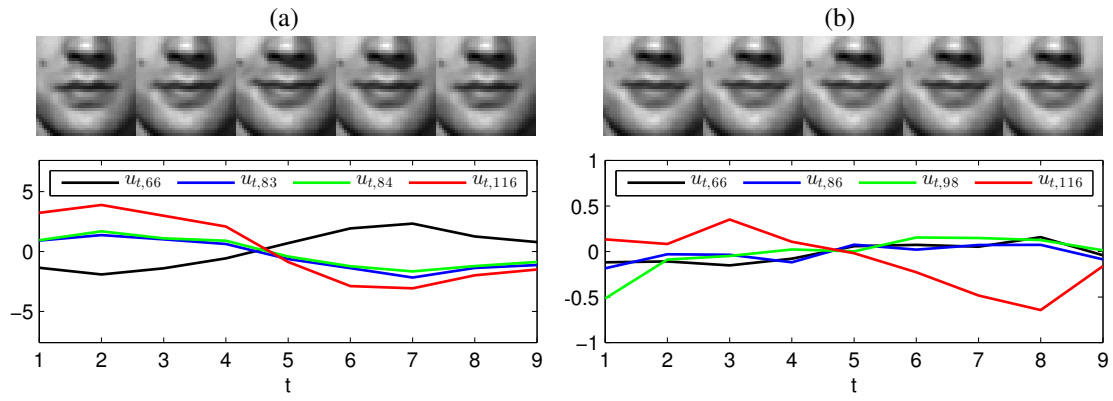
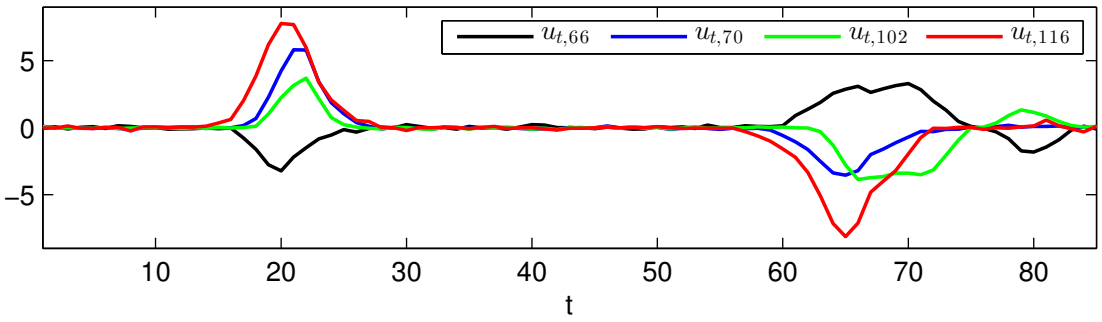


Figure 4.10: Two micro-expression sequences that contain a subtle lip corner movement. The movement in (b) is more subtle than the one in (a), hence the smaller coefficients (note that the y range of the latter plot is smaller). However, the basis A_{116} has the largest contribution in describing both sequences. The same basis is responsible also for describing the larger-intensity lip movements in Fig. 4.11–4.12

(a) A mouth sequence, representing the emotion of happiness.

(b) Coefficients $u_{t,k}$ computed from the entire sequence above jointly.



(c) Coefficients $u_{t,k}$ computed for disjoint segments of the sequence in (a) independently.

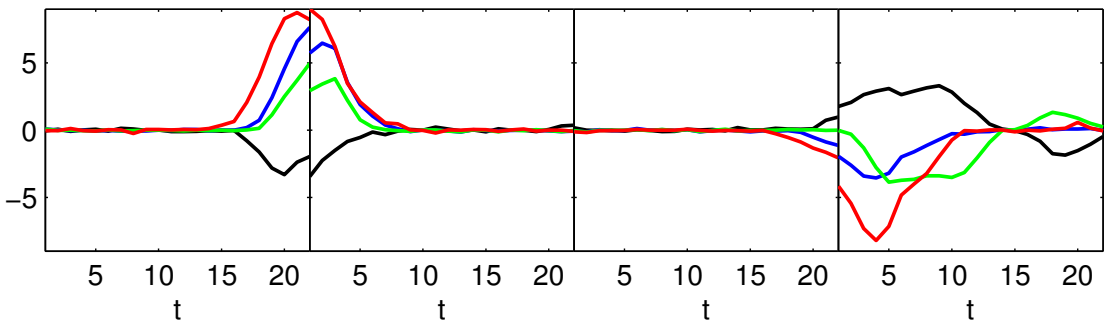
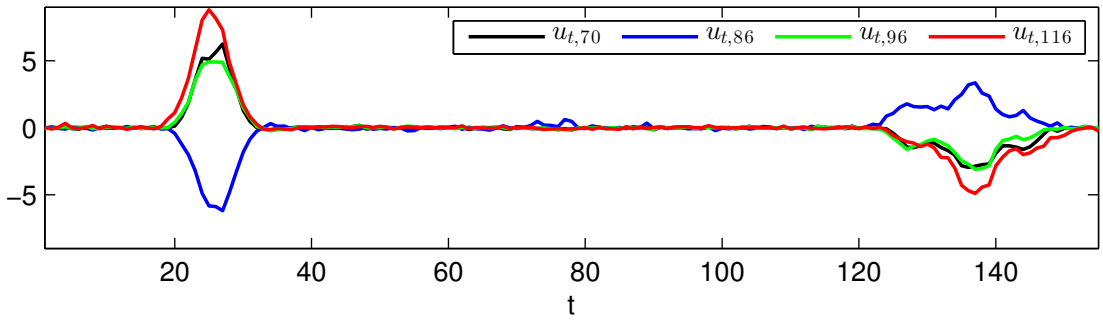


Figure 4.11: Illustration that depicts the coefficients $u_{t,k}$ computed from an exemplar sequence. For clarity, we depict only the coefficients $u_{t,k}$ obtained from the four most activated bases A_k.

(a) A mouth sequence, representing the emotion of happiness.

(b) Coefficients $u_{t,k}$ computed from the entire sequence above jointly.



(c) Coefficients $u_{t,k}$ computed for disjoint segments of the sequence in (a) independently.

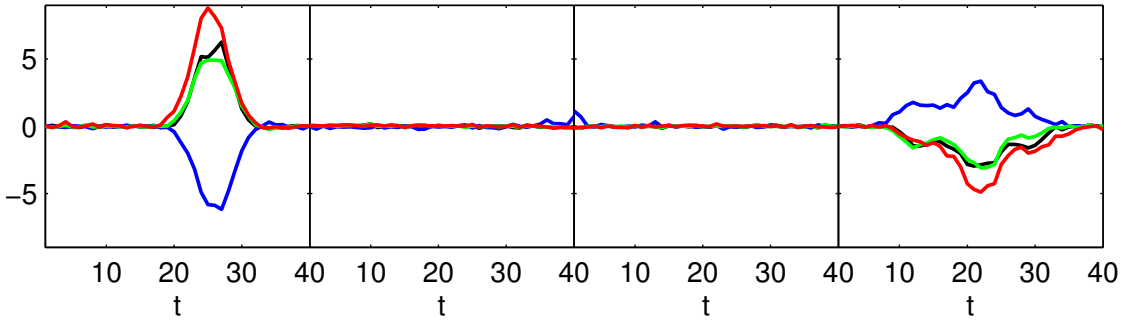


Figure 4.12: Illustration that depicts the coefficients $u_{t,k}$ computed from an exemplar sequence. For clarity, we depict only the coefficients $u_{t,k}$ obtained from the four most activated bases A_k.

(see the range of $t \in [40,60]$ in Fig.4.8b). This is an important ability for encoding naturalistic expressions, firstly because naturalistic expressions do not always follow the standard temporal phase order of neutral-onset-apex-offset (see Section 1.3). Moreover, even if the expression does follow this standard order, some of the phases of the expressions may not be visible. For example, the head pose may vary significantly between the onset and the offset, or there may be a partial occlusion or motion blurring in one of the phases.

In Appendix B.1 we visualise more coefficients that are computed from the six-basic emotion sequences of the first two MMI subjects.

4.8 Relationship with Slow Feature Analysis and Linear Dynamical Systems

In this section we discuss the similarities and differences between the proposed framework and two related approaches; slow feature analysis [256] (SFA) and a standard linear dynamical system

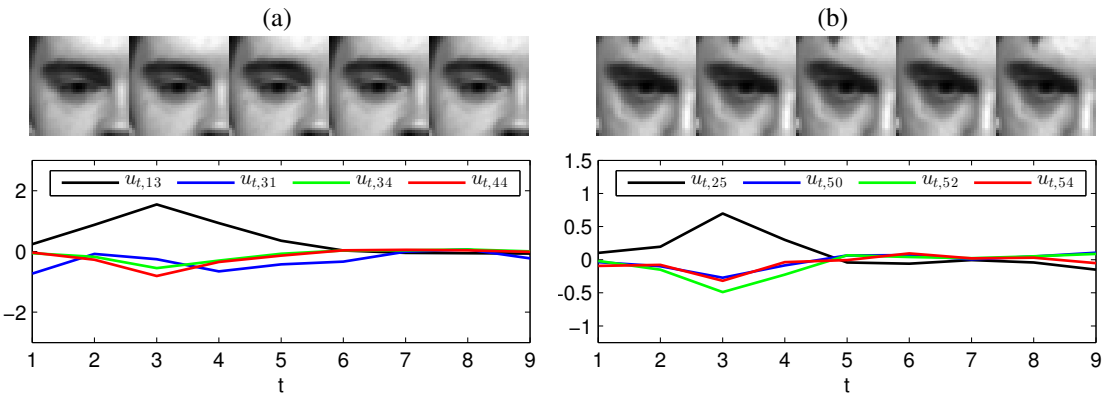


Figure 4.13: Two micro-expression sequences that contain subtle eyebrow movements.

(LDS).

The main similarity between our work and SFA is that we also add the slowness as a constraint as seen in (4.7). However, in our application slowness has a very specific role — to incorporate the prior knowledge that the speed of facial components changes gradually due to physical constraints [108]. In this regard, our slowness constraint is more akin to a classical temporal smoothness (or coherence) constraint that is employed when analysing motion [20]. On the contrary, SFA uses slowness for the much broader goal of extracting higher-order structures from data, and the slowness in SFA is not limited to motion. Below we briefly summarise SFA to compare it with our work further.

A video can be considered as data generated by latent variables; for example, in a video where individual objects enter and leave the visual scene one at a time, two latent variables can be considered to be the identity of the objects (i.e. "what" information) and their location (i.e. "where" information) [256]. Such a video explains well the motivation behind SFA [256]: Even though the raw sensory (i.e. pixel intensities) would generally undergo fast and abrupt changes, the latent variables undergo slow changes. For example, the "identity of the object" (*i.e.* "what") variable would be remain constant as long as the same object appears in the video, and the "location of the object" (*i.e.* "where") variable(s) would change gradually as the object enters and exits the scene. SFA aims at finding a transformation that would generate slowly changing features, with the motivation that those slow variables relate to the latent variables in the video. This aim is achieved by minimizing the (first-order) derivative of the learnt features and imposing three constraints. One of the constraints (the uncorrelated features constraint) provides SFA with

the property that the variables are ordered according to how slowly they change. This is the a key property for SFA; the slowest-changing (i.e. first) output can be considered to relate to the highest-order latent variable. For example, in an application to scene classification, the first variable can relate to the most important variable (*i.e.* the scene class) [222], and in application of AU phase detection, the first variable can directly relate to the phase of the AU [273].

On the contrary, in our framework there is no ordering in terms of the slowness of the learnt features (and consequently no constraint to impose an ordering). Each of the features has a pre-determined specific role: to reveal whether a particular type of motion took place. That is, even though each feature is different (it corresponds to a *different* type of motion), all of the features are conceptually same (they correspond to *a* particular type of motion). In contrast, in SFA features can be conceptually different and some may relate to motion while others not; for example, in an application to object recognition [256], some of the variable(s) would relate to the object identity, while others would relate to the object size or location with the aim of gaining invariance to such irrelevant factors. In our framework, the invariances that are considered are addressed prior to extracting the features $\mathbf{u}_t$; registration aims to eliminate size/location differences, and the Gabor phase shifts aim at partly reducing illumination variations.

Given the generality of the term LDS [19], the inference that takes place in (4.5) can also be considered as inference in an LDS. More specifically, the inferred features are related with a first-order Markov chain as (4.7) suggests. Inference in a standard LDS can be achieved using the Kalman filter equations (given that all the distributions are Gaussian) [19]; this would lead to a fully probabilistic approach and the benefits of adopting such an approach are very clear in some circumstances. For example, in a typical tracking problem the features that we aim to extract ($\mathbf{u}_t$) would correspond to the true location of an object whereas the Gabor phase shifts ($\dot{\psi}_t$) would correspond to (noisy) measurements. The Kalman equations would find at each time the optimal value $\mathbf{u}_t$ by considering the amount of estimated noise in the measurement $\dot{\psi}_t$ and the previous state $\mathbf{u}_{t-1}$. Our solution to (4.5), however, is deterministic. Our usage of a probabilistic framework is limited to arriving at the model (i.e. the a posteriori distribution $\log P(\mathcal{D}_{\dot{\psi}}|\mathbf{A})$) for learning the features; once we obtain this model, we convert it to an energy function which is solved deterministically. That is, we make point estimates of the features $\mathbf{u}_t$ without estimating the noise. Taking noise into account with a fully probabilistic approach can be beneficial in the cases where data can be noisy due, for example, to non-uniform illumination variations,

to large motion blur or to partial occlusions. However, a fully probabilistic approach is not straightforward for our model, as the Kalman equations cannot be used with the non-Gaussion distribution in (4.7) as the indefinite integral of (4.7) is intractable.

4.9 Automatic expression recognition with the bases

The learnt bases can be used to extract features for recognising the facial expression in a sequence $\mathbf{S}$. The features can be used as input to a multi-class classifier trained from a set of sequences, $\{\mathbf{S}^n\}_{n=1}^N$ (see Fig. 4.14).

The first step is computing the static, $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T)$, and dynamic, $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_T)$, basis coefficients for all frames of $\mathbf{S}$, as described in Section 4.5.2. Since the facial expression in $\mathbf{S}$ may not be temporally aligned with the training sequences $\{\mathbf{S}^n\}_{n=1}^N$, 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, $(\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^{T_A})$, where each slice $\mathbf{u}^\tau$ is a set that contains $Q_A = \lceil \frac{T}{T_A} \rceil$ coefficient vectors, *i.e.*:

$$\mathbf{u}^\tau = \{\mathbf{u}_{(\tau-1)Q_A+1}, \mathbf{u}_{(\tau-1)Q_A+2}, \ldots, \mathbf{u}_{\tau Q_A}\}. \tag{4.28}$$

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

$$\mathbf{h}^{\tau,k} = \text{hist}(\{u_{t',k} : u_{t',k} = [\mathbf{u}_{t'}]_k, \forall \mathbf{u}_{t'} \in \mathbf{u}^\tau\}), \tag{4.29}$$

where $\text{hist}(\cdot)$ 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 $\tau = 1, 2, \ldots, T_A$ and $k = 1, 2, \ldots, K_A$. 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, $(\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^{T_B})$. 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 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

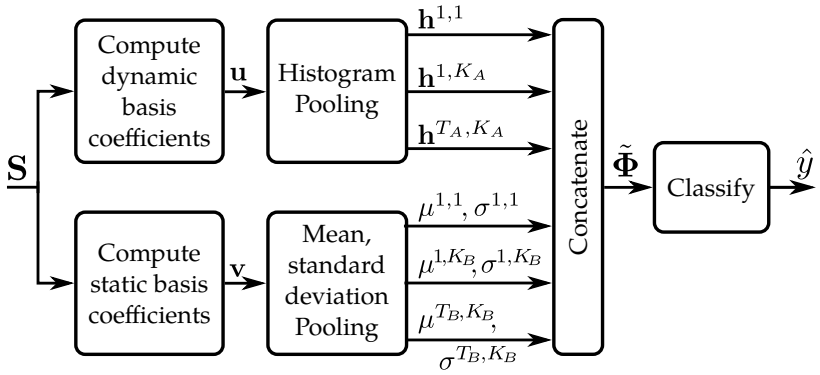


Figure 4.14: Block diagram of the proposed end-to-end process to predict $\hat{y}$, the expression in a sequence $\mathbf{S}$, with a pre-trained classifier.

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, $\tilde{\Phi}$, the feature vector of $\mathbf{S}$, 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.

4.10 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 4.6. 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[4] [261]. 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. 4.15a suggests, smaller squares may reduce performance on CK+ and MMI). Then we register temporally the cropped sequences with the MUMIE technique that we introduced in Chapter 3 in [189]. Finally, we re-scale the patches in the registered sequences to 32×32 pixels. As Fig. 4.15b shows, 32×32 achieves a good balance among the CK+, MMI and SMIC datasets.

We use a Gabor wavelet set with 4 orientations and 5 scales, which yields $D_W = 4468$ Gabor wavelets for frames of size 32×32. We use 4 orientations, instead of the more commonly used 8 [113], to reduce the dimensionality D_W, as the reconstruction performance with 4 and 8 orientations is similar on our images (see Fig. 4.16). The noise and prior parameters needed in Eq. (4.17–4.20) are set as $\tilde{\lambda}_v, \tilde{\lambda}_u = 10; \kappa = 4; \sigma_\rho = 0.25; \tilde{\lambda}_v, \tilde{\lambda}_u = 0.2$. We set $\tilde{\lambda}_v, \tilde{\lambda}_u, \kappa$ and σ_ρ

[4]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.

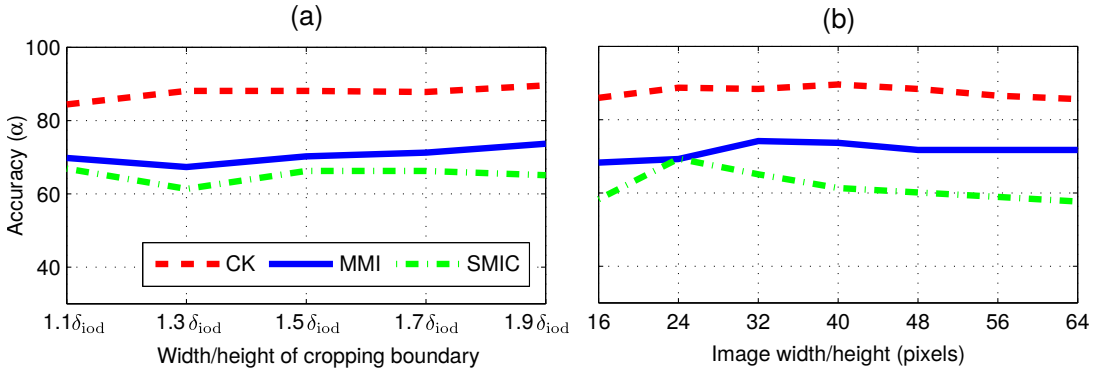


Figure 4.15: Performance variation with the dynamic bases with respect to (a) the size of the cropping rectangle in terms of inter-ocular distance, δ_{iod}, and (b) the size of the cropped patches after re-scaling ($K_A = 60$ for the MMI dataset and $K_A = 100$ for the CK+ and SMIC datasets).

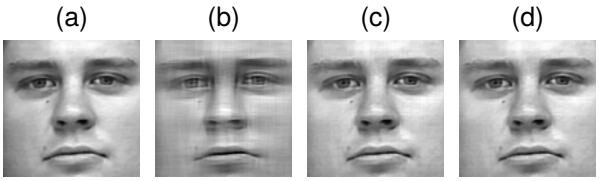


Figure 4.16: Reconstruction performance of sets that contain wavelets at 2, 4 and 8 different orientations. (a) Facial image from the CK+ dataset. (b), (c), and (d) show the reconstruction performance of wavelet sets that contain wavelets at 2, 4 and 8 orientations, respectively. Note that when increasing 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.

based on previous research [25], and $\tilde{\lambda}_v, \tilde{\lambda}_u$ 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 [198] and for the project$(\cdot)$ algorithm we use [81].

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 [114]) to 10 frames when learning the bases. Temporal interpolation effectively changes the frame rate of

the sequences. 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-*i*5 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 [281] representation on the same sequences is 0.023 seconds per frame.

4.11 Experiments

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.

4.11.1 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 4.2 and Fig. 4.17).

The CK+ dataset [96] 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.* LOSO cross validation [96]. 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 [167] 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

Table 4.2: 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

with an emotion. 205 sequences from 31 subjects fit these criteria. We also perform LOSO cross validation. The SMIC micro-expression dataset [114] 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) [114]. We use the data collected with a high-speed (100 fps) camera: 164 sequences with micro-expressions and 164 sequences with no micro-expressions.

4.11.2 Protocols

We use a C-SVM classifier with linear kernel [30] for CK+ and MMI tests by fixing the C parameter to 10^3 with no further optimisation. The baseline method in SMIC [114] 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:

$$\alpha = \frac{|\{y^n : y^n = \hat{y}^n\}_{n=1}^N|}{N}, \tag{4.30}$$

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.

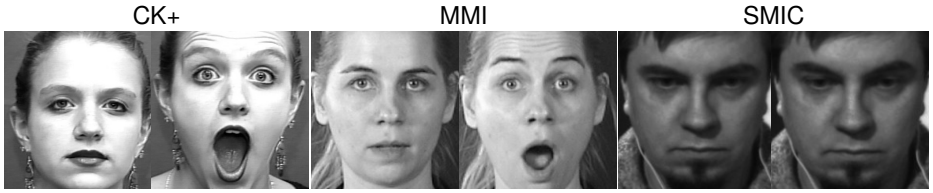


Figure 4.17: Examples from the CK+, MMI and SMIC datasets with a neutral frame and a frame with surprise expression, depicting that an emotion can be shown with expressions of different intensities. In the rightmost example, surprise is manifested with a subtle expression that involves an eyebrow movement.

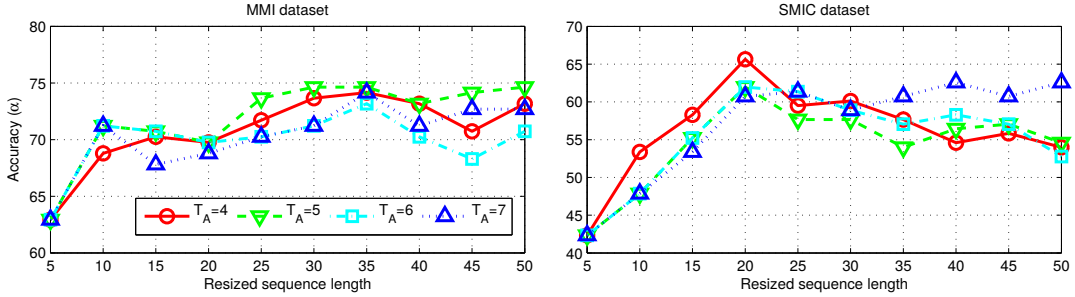


Figure 4.18: Performance with respect to (resized) sequence length T indicates sensitivity to frame rate, as the apparent motion speed changes when a sequence is resized temporally. Results are obtained with dynamic bases only.

4.11.3 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. 4.18 (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 4.10). 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 [114]. However, the performance of our method shows little variation for sequences

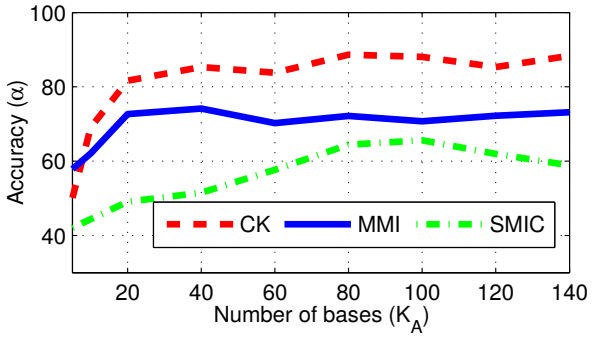


Figure 4.19: Performance of the dynamic features our method with respect to the number of bases K_A on the CK+, MMI and SMIC datasets.

Table 4.3: Performance of our method on CK+, MMI and SMIC, when bases are learnt in a within-database manner.

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. 4.19 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 4.3 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.* [194, 208]) 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 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 4.2).

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

Table 4.4: 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. †These results are obtained with a version of the Expressionlets method that requires supervised learning.

Ref.	Method	Needs Training Labels	Within-dataset validation	Cross-dataset validation	Accuracy on **CK+** (α)	Accuracy on **MMI** (α)	Accuracy on **SMIC** (α)
[85]	CFD-WL		N/A	N/A	92.32	–	–
[281], [125]	LBP-TOP		N/A	N/A	88.99	59.51	–
[100], [125]	3D-HOG		N/A	N/A	91.44	60.89	–
[202], [125]	3D-SIFT		N/A	N/A	81.35	64.39	–
[119]	Optical strain		N/A	N/A	–	–	53.56
[84]	STLBP-IP		N/A	N/A	–	–	57.93
[107]	AdaBst+STM		N/A	N/A	–	–	44.34
[281], [114]	LBP-TOP		N/A	N/A	–	–	49.30
[252]	ITBN		N/A	N/A	86.30	59.70	–
[93]	DTAGN	✓	✓		**96.94**	*66.33*	–
[124]	3DCNN-DAP	✓	✓		92.40	63.40	–
[125]	Expressionlets		✓		91.13	65.37	–
[125]	Expressionlets†	✓	✓		94.19	**75.12**	–
Proposed: FaceBases			✓		*96.02*	**75.12**	**65.64**
				✓	89.29	–	*60.36*

compare with on the CK+ and MMI datasets are Expressionlets [125], DTAGN [93] and 3DCNN-DAP [124] (see Table 2.3 for the extensions of the abbreviations). We further compare with Interval Temporal Bayesian Network (ITBN) [252], a method that proposes semantic modelling of expressions, as well as the (engineered) 3D-HOG [100], 3D-SIFT [202] and LBP-TOP [281] representations. We take the results reported in the papers.

Table 4.4 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 [114], Optical Strain [119], STLBP-IP [84] and a method that uses LBP-TOP with AdaBoost and Selective Transfer Machine (AdaBst+STM) [107]. 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 4.4), 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 ($\sim$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 4.2).

4.12 Limitations

The Gabor phase shifts that we use to encode motion can be sensitive to non-uniform illumination

variations within a sequence. Robustness against illumination variations can be improved at least in two ways: by replacing the representation or by adopting a probabilistic inference of the basis coefficients. As discussed in Section 4.4.3, the Gabor phase shifts representation can be replaced with recent optical flow methods that are capable of working on faces (i.e. capable of dealing with the potential absence of texture in facial skin). A probabilistic inference of the coefficients would allow the estimation of the noise on the images and thus find an optimal solution that weights past estimations and the current observation appropriately (see Section 4.8).

Since even part-based 2D registration cannot completely eliminate the motions caused by out-of-plane head rotations, applying the Facial Bases on data with such rotations may not yield a meaningful representation. Dealing with the head rotations requires training with a dataset that comprises various head poses, and a more sophisticated modelling that allows for non-frontal initial head poses (*e.g.* conditional modelling [214]) as well as head pose variations within sequences.

Moreover, we have observed some redundancy in the information provided by the bases: multiple bases can encode a similar facial movement, for example due to person-specific differences in facial appearance (see Section 4.6 and Section 4.7). This redundancy is a limitation of the proposed framework and it should ideally be eliminated not only to render the representation more compact, but also to allow for an easier semantic interpretation of the representation. That is, having a representation where one form of localised movement (*e.g.* eyebrow raising) corresponds to only one basis would be more desirable than having multiple bases.

The research conducted for other computer vision or human vision understanding problems provides fruitful future directions to reduce the redundancy of the representation proposed in this chapter. For example, one way to reduce redundancy is to add an additional layer to the framework that learns the relationships among bases (*e.g.* [97]), thus can identify the similarity of bases that encode a similar movement. An alternative approach is to use a model that can recognise the different transformations (*e.g.* spatial shift) of the same movement, such as a bilinear model [68].

Another limitation of the proposed representation is that for tractability we assumed independence between phase shifts and magnitudes and proposed to encode facial expressions with two distinct sets of bases — the static and the dynamic bases. However, phase shifts and magnitudes *are* dependent and a more accurate modelling requires to take this dependency into account. This

can be achieved by modelling phase shifts and magnitudes jointly, or by modelling dynamic bases conditioned on static bases.

4.13 Summary

In this chapter we proposed a novel dynamic representation for facial expression analysis that characterises 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.

Future directions for the proposed work are to reduce the identity bias, illumination and head pose sensitivity of the model as suggested in Section 4.12. Moreover, a natural extension of the proposed work is to use it for AU detection and AU phase recognition. Given the similarities between some of the learnt bases and the AUs (see Section 4.6), the proposed method can be tested for unsupervised detection for at least some of the AUs. Moreover, considering that the activation time of the coefficients typically corresponds to the onset and offset of the facial expressions (see Section 4.7 and Appendix B.1), the proposed method is promising in terms of recognising the temporal phases of the AUs in an unsupervised manner.

Chapter 5

Conclusion

5.1 Summary of findings and achievements

In this thesis we presented a comprehensive review of state-of-the-art affect analysers to identify the open issues and the practices that are common in successful systems. We addressed one of the open issues, namely, sequence registration, and proposed a robust facial sequence registration technique. We proposed a representation learning pipeline that represents facial expressions in terms of localised movements and is capable of recognising expressions across varying intensities.

In the literature review we presented in Chapter 2, we decomposed existing affect recognition systems into their fundamental components. In our summary (Section 2.8) we highlighted which system components help reducing sensitivity to illumination variations; we discussed the sensitivity of existing systems to registration errors; we outlined the ways in which head pose variations are addressed in the literature; we drew attention to the problem of identity bias; and we highlighted the benefits of combining multiple types of features such as appearance and shape representations.

Our summary in Section 2.8 also points to the increasing popularity of learning spatio-temporal representations from data, and to the potentials of such representations. However, we argue that learning a representation from time-varying facial data requires further validation to ensure that the representation is not sensitive to the frame rate, to the specific order of temporal phases, or to the intensity of expressions in the training sequences. We also argue that the appli-

cability of a representation pipeline to multiple models of emotion is a desired ability, and that the representations which are more suitable for some emotion models (*e.g.* continuous models) can be fundamentally different from those for other models (*e.g.* six-basic emotions).

We also argued that accurate registration is fundamental for the accurate analysis of facial motions, and developed a novel sequence registration technique that can be used for whole-face or part-based registration even under challenging illumination variations (Chapter 3). The summary of achievements and findings during the development of this technique are as follows.

- While rigid registration is a well-studied problem in computer vision, the existing methods that we used for evaluation have not been able to achieve reliable performance for part-based registration, *i.e.* when a facial feature (*e.g.* left eye, right eye or mouth) is cropped and registered independently from the rest of the sequence. Part-based registration is a difficult problem because the input sequence contains little texture/high-gradient regions and it undergoes non-rigid motions due to facial expressions.

- We proposed a novel registration framework that is based on computing motion locally with Gabor motion energy and then converting local motion into global motion with a set of pre-trained regressors.

- We computed the closed-form expressions of Gabor motion energy for a moving line and showed how to tune motion energy for a line moving with a specific speed and orientation.

- We showed that pre-trained regressors can accurately model the relationship between Gabor motion energy and rigid misalignment parameters, and that they can generalise and perform accurately on data with illumination variations even when trained using controlled data.

- We showed that drift errors that typify online registration can be reduced when the Gabor motion energy in the proposed framework is computed with respect to multiple frames.

Finally, in Chapter 5 we presented the proposed unsupervised representation learning framework. The summary of findings and achievements of this chapter are listed below.

- To the best of our knowledge, we presented the first learnt representation that mimics FACS by representing facial expressions in terms of localised facial movements and assigning an intensity-related coefficient to each movement.

- We showed that learning a generative linear model from Gabor phase shifts computed from facial videos produces basis functions that correspond to localised facial movements.

- With cross-database experiments we proved that the proposed framework has important generalisation abilities, as it achieves state-of-the-art performance even when the frame rate of the sequences varies by a factor of approximately 4, and even when a framework trained with posed and exaggerated expressions is used to recognise micro-expressions.

5.2 Limitations and future work

We discussed the limitations and future work for the methods proposed in this thesis separately for the registration framework (Section 3.8) and for the unsupervised representation learning scheme (Section 4.12), and here we provide a summary. The registration framework generally fails in the presence of large out-of-plane head pose variations, and the typical failure symptom is consecutive registration failures. In such cases it is suggested to restart registration by setting the most recent frame as the new reference for registration. Further future work for registration is to improve the computational efficiency that is compromised by the computationally complex spatio-temporal Gabor filters (see Section 3.8).

The limitations of the proposed unsupervised representation framework include sensitivity to out-of-plane head pose variations and to temporal illumination variations. Furthermore, visual analysis of the learnt bases suggests that some bases may suffer from identity bias and that multiple bases encode similar kinds of movement. As discussed in Section 4.12, the afore-listed sensitivities can be remedied by replacing the motion representation (*i.e.* Gabor phase shifts) with a more robust alternative, by adopting a probabilistic inference of the basis coefficients, and by employing a more complex modelling that can exploit the redundancy among different bases. A natural future direction for the proposed method is to apply it for AU detection and AU phase recognition in a supervised or unsupervised manner (Section 4.12).

5.3 Closing remarks and outlook

A few years ago two of the major research directions identified for facial affect recognition were [276]: (i) the recognition of affect "in-the-wild" and (ii) the recognition of subtle expressions.

In today's literature, the former of the two issues is being addressed increasingly more – at

least by spatial affect analysis pipelines – thanks to the development of robust facial landmark localisation techniques and the hierarchical representation pipelines such as deep architectures. However, such systems have generally been validated for relatively large-intensity (*i.e.* pronounced) facial expressions and for the six-basic emotions model. One problem is that the six-basic emotions are limited in their ability to represent everyday emotions [73]; therefore, even a system that achieves perfect accuracy may have a limited relevance for a real-life application. While continuous models are a promising alternative for representing daily-life emotions (see Section 1.3), efforts to collect data "in-the-wild" and annotate it with continuous models started only recently [275].

This brings us to the second of the above-mentioned research directions. Identifying subtle expressions, which are associated with low-intensity emotions, was one of the most important motivations to employ spatio-temporal representations instead of the simpler spatial representations (see Section 2.5.2). The direction taken by the literature in this regard has been rather unexpected, if not puzzling. Sophisticated data-driven spatio-temporal representations have been developed recently; however, most of them have not exploited the capability of a spatio-temporal approach to recognise subtle expressions.

An outstanding issue in today's literature is to devise systems that are capable of operating "in-the-wild" and also identifying subtle expressions. Hierarchical spatio-temporal representations devised through machine learning have the potential to address those two issues concurrently, as hierarchical representations encourage robustness and spatio-temporal representations facilitate the analysis of subtle expressions. With the overwhelming research on sophisticated deep architectures [110] and with tech giants such as Microsoft [52] and Google [80] investing in emotion recognition, the progress can happen at an unexpectedly fast rate.

This thesis hopes to have contributed to this progress threefold. Firstly, we aimed to underpin useful practices in the design *and* the evaluation of data-driven spatio-temporal approaches (Section 2.8). Secondly, we drew attention to the importance accurate registration for analysing subtle expressions and proposed a sequence registration technique (Chapter 3). Finally, we proposed a novel spatio-temporal representation for facial analysis (Chapter 4). This representation is based on the simple idea of describing facial expressions in terms of localised movements and discerning between the different intensities of the same movement. While the representation proposed in this thesis would be called shallow in today's literature, this simple idea that lead to its

success is implemented through localised differential filters that are commonplace in hierarchical representations, therefore can be incorporated into novel shallow *or* deep architectures.

Appendices

Appendix A

Gabor motion energy

A.1 Computing motion energy of a moving line

In this section we compute the energy of a moving line, which will enable us to study the properties of motion energy analytically. For brevity, we denote the moving line with $\mathbf{I}$ (instead of $\mathbf{I}_l$); and the even- and odd-phased Gabor filters respectively with g^e and g^o. To compute the energy of the moving line, $E_{\mathbf{I}_l}$, we must compute the convolutions $\mathbf{I} * g^e$ and $\mathbf{I} * g^o$. The convolution $\mathbf{I} * g^e$ requires the computation a triple integral that can be challenging even for a computer algebra system. Therefore we make use of the Convolution Theorem, which states that, under suitable conditions, the convolution of two functions is equivalent to the pointwise product of their Fourier transforms, *i.e.* $\mathcal{F}\{\mathbf{I} * g\} = \mathcal{F}\{\mathbf{I}\}\mathcal{F}\{g\}$. The Fourier transforms of $\mathbf{I}$, g^e are denoted respectively with $\hat{\mathbf{I}}$, $\hat{g^e}$ and are computed using Mathematica® as (see Appendix C.1):

$$\hat{\mathbf{I}}(\xi_1, \xi_2, \xi_3) = c\frac{\sqrt{2\pi}}{\cos\theta_l}\,\delta\left(\xi_1 v_l \sec\theta_l + \xi_3\right) \delta(\xi_1 \tan\theta_l + \xi_2), \tag{A.1}$$

$$\hat{g^e}(\xi_1, \xi_2, \xi_3) = \frac{1}{4\sqrt{2}\pi\sqrt{\pi}}\left(1 + e^{\xi_3 v_g + \xi_1 \cos\theta_g + \xi_2 \sin\theta_g}\right) e^{-\frac{2\xi_2 \sin\theta_g + 1 + \xi_1^2 + \xi_2^2 + (\xi_3 + v_g)^2 + 2\xi_1 \cos\theta_g}{4}}. \tag{A.2}$$

$\mathbf{I} * g^e$ is obtained with an inverse transform, $\mathbf{I} * g^e = \mathcal{F}^{-1}\{\hat{\mathbf{I}}\hat{g^e}\}$ (see Appendix C.1 and Appendix C.2):

$$\begin{aligned}\mathbf{I} * g^e = & \frac{c\,\mathrm{sgn}(\sec\theta_l)}{2\sqrt{2}\pi^2\sqrt{1+v_l^2}} \\ & \cos\frac{(v_g v_l - \cos\theta_{gl})(tv_l - x\cos\theta_l + y\sin\theta_l)}{1+v_l^2} \\ & e^{-\frac{v_g v_l\cos\theta_{gl} + 4tyv_l\sin\theta_l - 4x\cos\theta_l(tv_l+\sin\theta_l)}{2(1+v_l^2)}} \\ & e^{-\frac{1+4x^2+4y^2+2v_g^2+(2+8t^2)v_l^2-\cos 2\theta_{gl}+4(x^2-y^2)\cos 2\theta_l}{8(1+v_l^2)}},\end{aligned} \tag{A.3}$$

where $\theta_{gl} \triangleq \theta_g + \theta_l$. The convolution with the odd-phased filter, $\mathbf{I} * g^o$, produces a similar output — the only difference is that the first cos function is replaced with $-\sin$. Finally, using $\mathbf{I} * g^e$ and $\mathbf{I} * g^o$, we can compute the energy for the moving line $E_\mathbf{I} = (\mathbf{I} * g^e)^2 + (\mathbf{I} * g^o)^2$ as:

$$\begin{aligned}E_\mathbf{I} = & \frac{\bar{c}^2}{1+v_l^2} e^{-\frac{v_g v_l\cos\theta_{gl} + 4tyv_l\sin\theta_l - 4x\cos\theta_l(tv_l+y\sin\theta_l)}{1+v_l^2}} \\ & e^{-\frac{1+4x^2+4y^2+2v_g^2+(2+8t^2)v_l^2-\cos 2\theta_{gl}+4(x^2-y^2)\cos 2\theta_l}{4(1+v_l^2)}},\end{aligned} \tag{A.4}$$

where $\bar{c} \triangleq \frac{c}{2\sqrt{2}\pi^2}$. An interactive plot that shows how $E_\mathbf{I}$ varies with filter parameters θ_g, v_g and line parameters θ_l, v_l is provided in `ftp://spit.eecs.qmul.ac.uk/pub/es/s.zip`.

A.2 Tuning direction and velocity for Gabor motion energy

In order to tune a Gabor filter pair to a particular speed v_l and spatial orientation θ_l, we should find the v_g and θ_g values that maximise $E_\mathbf{I}$. To this end, we first find all extrema of $E_\mathbf{I}$, and then find which of these are the maxima.

To find extrema, we compute the first-order partial derivatives of $E_\mathbf{I}$ with respect to v_g and θ_g:

$$\frac{\partial E_\mathbf{I}}{\partial v_g} = -\frac{1}{1+v_l^2}(v_g + v_l\cos\theta_{gl})E_\mathbf{I}, \tag{A.5}$$

$$\frac{\partial E_\mathbf{I}}{\partial \theta_g} = -\frac{1}{1+v_l^2}(\cos\theta_{gl} - v_g v_l)\sin\theta_{gl} E_\mathbf{I}. \tag{A.6}$$

The solutions that make both partial derivatives zero can be considered as four sets, S_1, S_2, S_3, S_4,

that are defined as:

$$S_1 \triangleq \{(v_g,\theta_g) : (0,-\pi/2-\theta_l+2\pi k), k\in\mathbb{Z}\}, \tag{A.7}$$

$$S_2 \triangleq \{(v_g,\theta_g) : (0,\pi/2-\theta_l+2\pi k), k\in\mathbb{Z}\}, \tag{A.8}$$

$$S_3 \triangleq \{(v_g,\theta_g) : (v_l,\pi-\theta_l+2\pi k), k\in\mathbb{Z}\}, \tag{A.9}$$

$$S_4 \triangleq \{(v_g,\theta_g) : (-v_l,-\theta_l+2\pi k), k\in\mathbb{Z}\}. \tag{A.10}$$

We eliminate S_4 as we assume $v_l, v_g \geq 0$, so the only solution to satisfy S_4 is $v_l = v_g = 0$, which implies that the line is not moving. To determine whether there is a maximum among the remaining solutions, S_1, S_2, S_3, we use the second derivative test. The second partial derivatives of $E_{\mathbf{I}}$ are:

$$\frac{\partial E_{\mathbf{I}}^2}{\partial^2 v_g} = \left[v_l\cos\theta_{gl}(2v_g+v_l\cos\theta_{gl}) - 1 + v_g^2 - v_l^2\right]\frac{E_{\mathbf{I}}}{(1+v_l^2)^2}, \tag{A.11}$$

$$\frac{\partial E_{\mathbf{I}}^2}{\partial^2 \theta_g} = \left[(1+v_l^2)(v_g v_l\cos\theta_{gl} - \cos 2\theta_g l) + (\cos\theta_{gl} - v_g v_l)^2\sin^2\theta_{gl}\right]\frac{E_{\mathbf{I}}}{(1+v_l^2)^2}, \tag{A.12}$$

$$\frac{\partial E_{\mathbf{I}}^2}{\partial\theta_g \partial v_g} = \left[v_l(3-2v_g^2+2v_l^2+\cos 2\theta_{gl}) - 2v_g\cos\theta_{gl}(v_l^2-1)\right]\frac{E_{\mathbf{I}}\sin\theta_{gl}}{2(1+v_l^2)^2}. \tag{A.13}$$

To perform the second partial derivative test, we construct the Hessian matrix H and compute its determinant as a function $D(v_g,\theta_g)$ as follows:

$$H = \begin{bmatrix} \frac{\partial E_{\mathbf{I}}^2}{\partial^2 v_g} & \frac{\partial E_{\mathbf{I}}^2}{\partial v_g \partial\theta_g} \\ \frac{\partial E_{\mathbf{I}}^2}{\partial\theta_g\partial v_g} & \frac{\partial E_{\mathbf{I}}^2}{\partial^2\theta_g} \end{bmatrix}, \tag{A.14}$$

$$D(v_g,\theta_g) \triangleq \det(H) = \frac{\partial E_{\mathbf{I}}^2}{\partial^2 v_g}\frac{\partial E_{\mathbf{I}}^2}{\partial^2\theta_g} - \left(\frac{\partial E_{\mathbf{I}}^2}{\partial\theta_g\partial v_g}\right)^2. \tag{A.15}$$

To determine whether the solutions S_1, S_2 or S_3 are extrema, we denote the determinants of those solutions respectively as $D_{S_1}, D_{S_2}, D_{S_3}$ and compute them as (see Appendix C.3):

$$D_{S_1} = D_{S_2} = -Ke^{\frac{2(y^2-x^2)\cos 2\theta_l - 8tyv_l\sin\theta_l + 4xy\sin 2\theta_l}{1+v_l^2}} e^{-\frac{8txv_l\cos\theta_l - 1 - 2x^2 - 2y^2 - v_l^2 - 4t^2v_l^2}{1+v_l^2}}, \tag{A.16}$$

$$D_{S_3} = Ke^{-\frac{4}{1+v_l^2}(tv_l - x\cos\theta_l + y\sin\theta_l)^2}, \tag{A.17}$$

where $K = \frac{c^2}{(2\sqrt{2}\pi^2)^4(1+v_l^2)^3} > 0$. Since the outcome of the exp function is always positive, D_{S_1}, D_{S_2} are always negative; therefore, S_1, S_2 contain saddle points and not extrema. On the other hand, S_3 contains extrema as $D_{S_3} > 0$. To check whether S_3 contains maxima or minima, we check the

partial derivative $\frac{\partial E_{\mathbf{I}}^2}{\partial^2 v_g}$ for the solutions of S_3:

$$\left.\frac{\partial E_{\mathbf{I}}^2}{\partial^2 v_g}\right|_{(v_g,\theta_g)\in S_3} = \frac{-c\,e^{-\frac{2(y\sin\theta_l + tv_l - x\cos\theta_l)^2}{1+v_l^2}}}{8\pi^4(1+v_l^2)^2}. \tag{A.18}$$

This expression is always negative, therefore $(v_g,\theta_g)\in S_3$ are maxima. In conclusion, to tune the filters g^e, g^o to a line moving with spatial orientation θ_l and speed v_l, the filter parameters v_g and θ_g must be defined as follows:

$$v_g = v_l, \tag{A.19}$$

$$\theta_g = \pi - \theta_l + 2\pi k. \tag{A.20}$$

A.3 Variation of normalisation coefficients against time

We show that the $Z_{\mathbf{I}_p^w}$ coefficient that appears in (3.18) after illumination is cancelled out changes slowly with time w, and therefore causes little variation in the trend of the signal we aim to measure (*i.e.* motion energy). This is important, as it ensures that during normalisation we are not altering the characteristic behaviour of motion energy, which was discussed throughout Section 3.4. We analyse the sensitivity of $Z_{\mathbf{I}^t}$ to t on a sequence $\mathbf{I}$ where there is a global translation and no illumination variations — a sequence that adheres to the definition of $\mathbf{I}_p$ in Section 3.4.2. We may show that $Z_{\mathbf{I}^t}$ varies slowly with time by showing that the ℓ_1 distance between the coefficients of two frames, $|Z_{\mathbf{I}^{t_m}} - Z_{\mathbf{I}^{t_n}}|$, is small. To compute $Z_{\mathbf{I}^{t_m}}, Z_{\mathbf{I}^{t_n}}$, we first need to obtain the static sequences $\mathbf{I}^{t_m}, \mathbf{I}^{t_n}$. Since the only difference between the frames of $\mathbf{I}$ is a global translation, the static sequences are translated versions of each other (similarly to Fig. 3.5b,c), that is, $\mathbf{I}^{t_m}(\mathbf{x}) = \mathbf{I}^{t_n}(\mathbf{x}+\tau)$ for some $\tau = (\tau_x, \tau_y, 0)$.

To compute $Z_{\mathbf{I}^{t_m}}$ as in (3.11), we first need to compute the energy, which is based on convolution. Let $h_n(\mathbf{x},g) \triangleq (\mathbf{I}^{t_n} * g)(\mathbf{x})$ and $h_m(\mathbf{x},g) \triangleq (\mathbf{I}^{t_m} * g)(\mathbf{x})$. Since translation commutes with convolution, h_m can be rewritten as:

$$h_m(\mathbf{x},g) = h_n(\mathbf{x}+\tau, g). \tag{A.21}$$

The energies $E_{\mathbf{I}^{t_n}}(\mathbf{x})$ and $E_{\mathbf{I}^{t_m}}(\mathbf{x})$ can then be computed as:

$$E_{\mathbf{I}^{t_n}}(\mathbf{x}) = (h_n(\mathbf{x}, g^e))^2 + (h_n(\mathbf{x}, g^o))^2, \tag{A.22}$$

$$\begin{aligned} E_{\mathbf{I}^{t_m}}(\mathbf{x}) &= (h_m(\mathbf{x}, g^e))^2 + (h_m(\mathbf{x}, g^o))^2 \\ &= (h_n(\mathbf{x}+\tau, g^e))^2 + (h_n(\mathbf{x}+\tau, g^o))^2 \\ &= E_{\mathbf{I}^{t_n}}(\mathbf{x}+\tau). \end{aligned} \tag{A.23}$$

Then, for a volume $\Omega \triangleq X \times Y \times T \triangleq (x_0, x_f) \times (y_0, y_f) \times (t_0, t_f)$, the coefficients $Z_{\mathbf{I}^{t_n}}$, $Z_{\mathbf{I}^{t_m}}$ can be computed as:

$$Z_{\mathbf{I}^{t_n}} = \int_{\Omega} E_{\mathbf{I}^{t_n}}(\mathbf{x}')\mathrm{d}\mathbf{x}', \tag{A.24}$$

$$Z_{\mathbf{I}^{t_m}} = \int_{\Omega} E_{\mathbf{I}^{t_m}}(\mathbf{x}')\mathrm{d}\mathbf{x}' = \int_{\Omega} E_{\mathbf{I}^{t_n}}(\mathbf{x}'+\tau)\mathrm{d}\mathbf{x}'. \tag{A.25}$$

The distance $|Z_{\mathbf{I}^{t_n}} - Z_{\mathbf{I}^{t_m}}|$ can rewritten with a change of variable in the integral of $Z_{\mathbf{I}^{t_m}}$. Let $X' \triangleq (x_0+\tau_x, x_f+\tau_x)$, $Y' \triangleq (y_0+\tau_y, y_f+\tau_y)$ and $\Omega' \triangleq X' \times Y' \times T$. Then, it can be shown that:

$$|Z_{\mathbf{I}^{t_n}} - Z_{\mathbf{I}^{t_m}}| = \left| \int_{\Omega} E_{\mathbf{I}^{t_n}}(\mathbf{x}')\mathrm{d}\mathbf{x}' - \int_{\Omega'} E_{\mathbf{I}^{t_n}}(\mathbf{x}')\mathrm{d}\mathbf{x}' \right|. \tag{A.26}$$

We can interpret (A.26) better by excluding the region of intersection, $\Omega \cap \Omega'$: This region will have no contribution to the distance in (A.26), as the integrands of the two integrals in (A.26) are equal, and their difference would yield zero when the integration is done over the same region. The non-zero contribution to (A.26) can only come from the non-intersecting regions: $\Omega \backslash \Omega'$ and $\Omega' \backslash \Omega$. These regions depend on the amount of translation: if translation is small, then $\Omega \backslash \Omega', \Omega' \backslash \Omega$ become small, and therefore $|Z_{\mathbf{I}^{t_n}} - Z_{\mathbf{I}^{t_m}}|$ is likely to be small.

In Fig. A.1 we show quantitatively how two successive coefficients $Z_{\mathbf{I}^0}$, $Z_{\mathbf{I}^1}$ change with respect to the amount of translation. To this end, we crop $N = 1000$ image samples, $\{I_n\}_{n=1}^N$, each of size $2P \times 2P$, from randomly picked regions in the first frames of the test sequences. We synthesize two-frame sequences such as $\mathbf{I}_{n,\tau} = (I_n, I'_n)$, where I_n denotes a sample and I'_n denotes the same sample after being translated horizontally by τ pixels. We synthesise 10 sequences per sample, $\mathbf{I}_{n,\tau_0}, \mathbf{I}_{n,\tau_1}, \ldots, \mathbf{I}_{n,\tau_9}$, such as $\tau_i = i/2$. We can measure how $Z_{\mathbf{I}^t_{n,\tau}}$ varies between the two frames of $\mathbf{I}_{n,\tau}$ through the ratio $Z_{\mathbf{I}^1_{n,\tau}}/Z_{\mathbf{I}^0_{n,\tau}}$. Specifically, we use the average of this ratio over all sequences,

$$\delta\bar{Z}_\tau \triangleq \sum_{n=1}^{N} \frac{Z_{\mathbf{I}^1_{n,\tau}}}{Z_{\mathbf{I}^0_{n,\tau}}}. \tag{A.27}$$

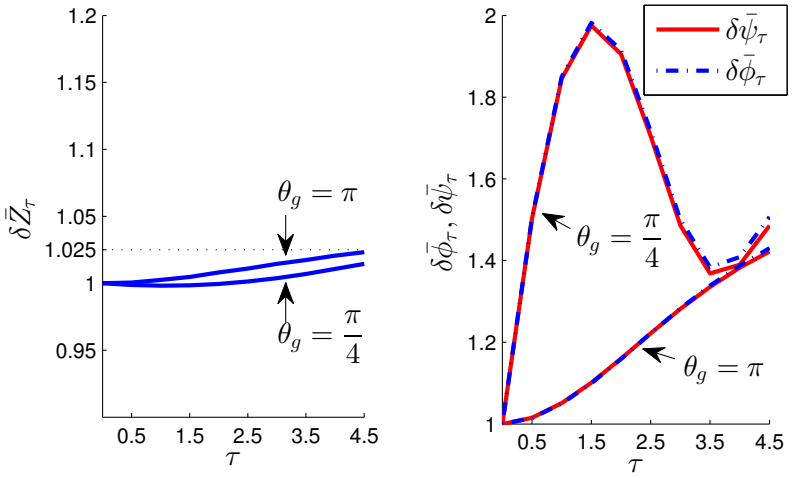


Figure A.1: Illustration which depicts that the $Z_{\mathbf{I}^t}$ coefficient shows small variation over time (left), and that this variation causes a negligible change in the trend of the motion energy function. Results are obtained with two pairs of filters tuned to different orientations θ_g but to a common speed $v_g = 1$.

In Fig. A.1 (left) we show how $\delta\bar{Z}_\tau$ changes with τ, for coefficients computed with two filter pairs tuned to different orientations. We note that $\delta\bar{Z}_\tau$ generally deviates from 1 proportionally to τ, which is in accordance with the conclusion we reached after (A.26). However, this increase is relatively small; the deviation in $\delta\bar{Z}_\tau$ never exceeds 2.5%, which shows that $Z_{\mathbf{I}^t_{n,\tau}}$ has little sensitivity to amount of translation, τ.

We now analyse whether this increase is significant: We illustrate how normalisation with the time-dependent $Z_{\mathbf{I}^t}$ coefficients changes the trend of the signal that we aim to measure — the Gabor motion energy. For this purpose, we compute how the pooling output of the sequences varies with τ:

$$\delta\bar{\phi}_\tau \triangleq \sum_{n=1}^{N} \frac{\phi_{n,\tau}}{\phi_{n,\tau_0}}, \tag{A.28}$$

where $\phi_{n,\tau}$ is the output of mean pooling of the normalised energy of $\mathbf{I}_{n,\tau}$. We compare $\delta\bar{\phi}_\tau$ with the original (*i.e.* un-normalised) energy, by computing the following ratio:

$$\delta\bar{\psi}_\tau \triangleq \sum_{n=1}^{N} \frac{\psi_{n,\tau}}{\psi_{n,\tau_0}}, \tag{A.29}$$

where $\psi_{n,\tau}$ denotes a pooling output computed from the un-normalised energy. Note that $\delta\bar{\phi}_\tau$ and $\delta\bar{\psi}_\tau$ can be compared fairly, because both are divided with the pooling output of the non-moving sequence. Ideally, we would like $\delta\bar{\psi}_\tau$ and $\delta\bar{\phi}_\tau$ to be the same for any τ value.

Finally, in Fig. A.1 (right) we compare $\delta\bar{\phi}_\tau$ with $\delta\bar{\psi}_\tau$: The difference between $\delta\bar{\phi}_\tau$ and $\delta\bar{\psi}_\tau$ is small even for the largest τ value. It is therefore reasonable to assume that the $Z_{\mathbf{I}^t}$ coefficients

cause a negligible change in the trend of the energy, given that the sequence they are computed from contains no illumination variations.

A.4 Efficient computation of normalised motion energy

Applying the normalisation proposed in Section 3.4.2 locally can be computationally involved, because each local region requires its own normalisation coefficients. In this section we show how the summed area tables [42] can be used to perform pooling of Gabor motion energy efficiently. Let us consider the computation of mean pooling of energy over a $P \times P$-sized region centred anchored at the pixel (i, j):

$$\phi^{ij} = \frac{1}{P^2} \sum_{x,y=\delta P+1}^{P+\delta P} E_{\tilde{\mathbf{I}}_{ij}}[x,y], \tag{A.30}$$

where with $\mathbf{I}_{ij} \triangleq (I_{ij,0}, I_{ij,1})$ we denote the part of the image sequence that the local energy will be computed from; that is, $I_{ij,t}$ is the $P \times P$-sized region of the image I_t centred on (i, j).

We first compute the convolutions required for energy on the entire input images and then use the necessary values for ϕ^{ij}. Note that for a filter g, the equality $(I_{ij,t} * g)[x,y] = (I_t * g)[x+i, y+j]$ holds due to our definition of $I_{ij,t}$ and the fact that translation commutes with convolution. Let A_t^{xy}, B_t^{xy} be $A_t^{xy} \triangleq (I_t * g^e_{T_g-1-t})[x,y]$ and $B_t^{xy} \triangleq (I_t * g^o_{T_g-1-t})[x,y]$. Using the afore-mentioned equality, we can perform the pooling in (A.30) as:

$$\begin{aligned}
\phi^{ij} =& \frac{1}{P^2} \sum_{x,y} \left[\left(\sum_{t=0}^{T_g-1} \frac{1}{\sqrt{Z_{\mathbf{I}^t_{ij}}}} A_t^{xy} \right)^2 + \left(\sum_{t=0}^{T_g-1} \frac{1}{\sqrt{Z_{\mathbf{I}^t_{ij}}}} B_t^{xy} \right)^2 \right] \\
=& \frac{\sum_{x,y}(A_0^{xy})^2}{P^2 Z_{\mathbf{I}^0_{ij}}} + \frac{\sum_{x,y}(A_1^{xy})^2}{P^2 Z_{\mathbf{I}^1_{ij}}} + \frac{\sum_{x,y}(B_0^{xy})^2}{P^2 Z_{\mathbf{I}^0_{ij}}} + \frac{\sum_{x,y}(B_1^{xy})^2}{P^2 Z_{\mathbf{I}^1_{ij}}} \\
&+ \frac{2}{P^2 \sqrt{Z_{\mathbf{I}^0_{ij}} Z_{\mathbf{I}^1_{ij}}}} \left(\sum_{x,y} A_0^{xy} A_1^{xy} + \sum_{x,y} B_0^{xy} B_1^{xy} \right),
\end{aligned} \tag{A.31}$$

where we dropped the dependence of ϕ_k^{ij} to k for clarity. The the sums in the right-hand-side run over $(x,y) \in \mathbb{N}_{[i+1,i+P]} \times \mathbb{N}_{[j+1,j+P]}$. After writing the sums as in (A.31), we can employ summed-area tables, which enable the computation of each sum with four instead of P^2 operations [42]. The integrals (*i.e.* sums) required for $Z_{\mathbf{I}^0_{ij}}, Z_{\mathbf{I}^1_{ij}}$ can also be computed in a similar manner, once the summed-area table of the static energies, $E_{\mathbf{I}^0}, E_{\mathbf{I}^1}$, where $\mathbf{I} \triangleq (I_0, I_1)$, are pre-computed.

Appendix B

Additional illustrations

B.1 Basis coefficients from six-basic emotions

This section visualises the basis coefficients $u_{t,k}$ (see Section 4.7) computed from the sequences of the first two subjects of the MMI dataset [167]. Fig. B.1–B.6 display the anger, disgust, fear, happiness, sadness, surprise sequences for Subject 001, and Fig. B.7–B.12 respectively display the sequences of those emotions for Subject 002. Each figure can be split vertically into two sub-figures: one for the left eye and one for the mouth. Each of those sub-figures is constructed in the same way as the corresponding figures in Section 4.7 (*e.g.* see Fig. 4.8); that is, we split the sub-figure into three parts where the top part represents the input sequence, the middle part represents the coefficients computed from the entire sequence jointly, and the bottom part represents the coefficients computed from four disjoint segments of the sequence independently.

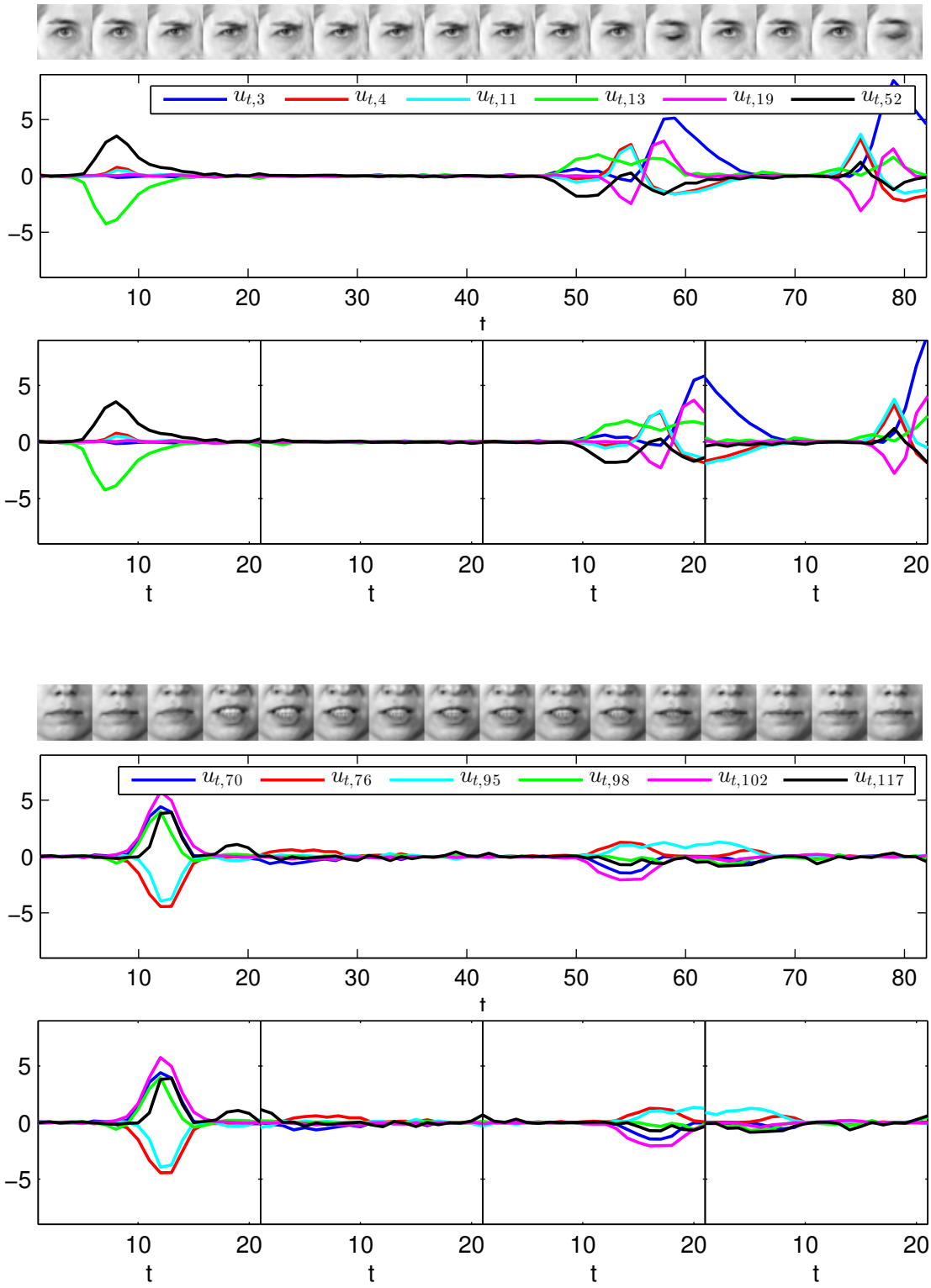


Figure B.1: Subject 1, expression of anger.

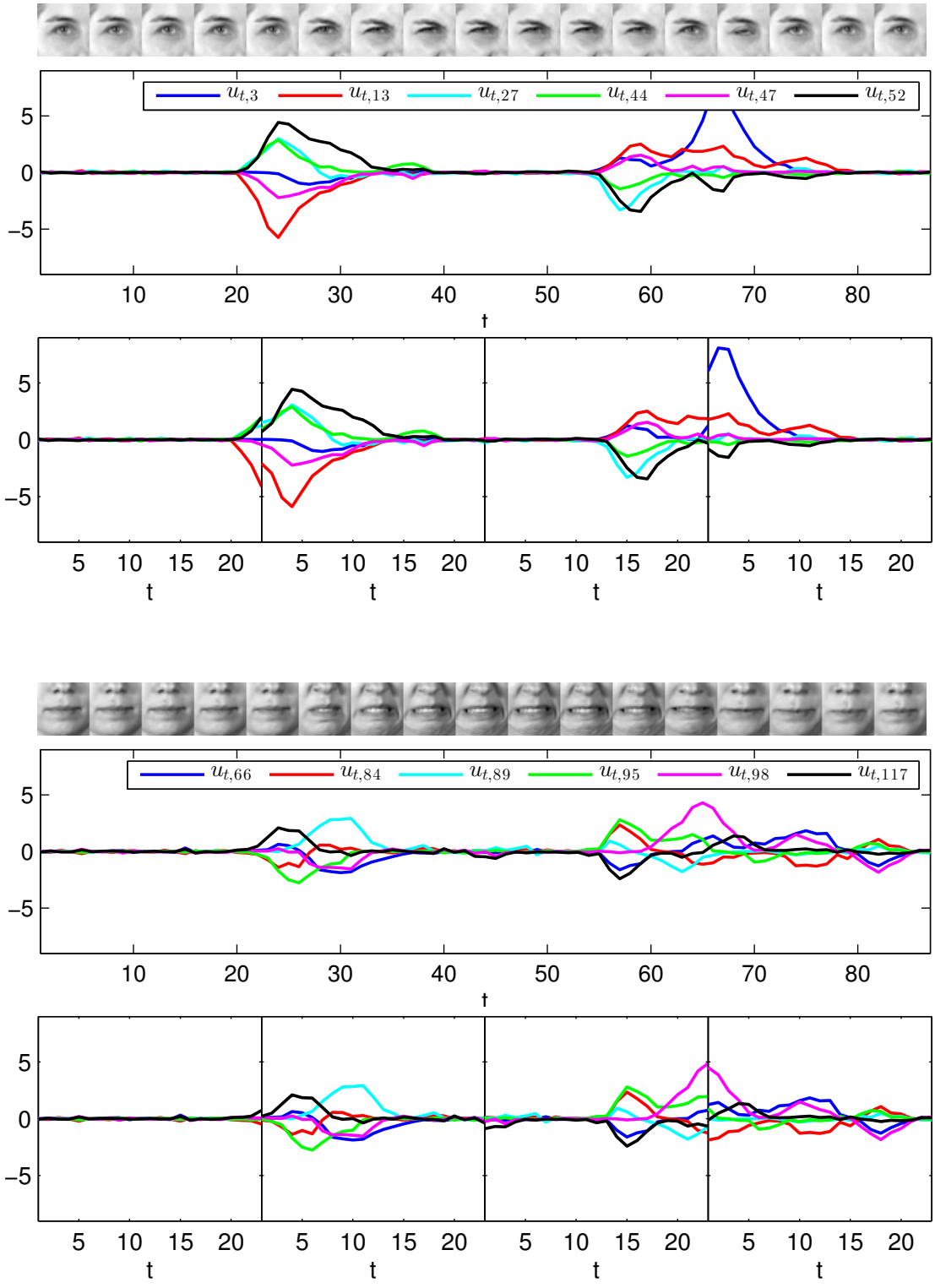


Figure B.2: Subject 1, expression of disgust.

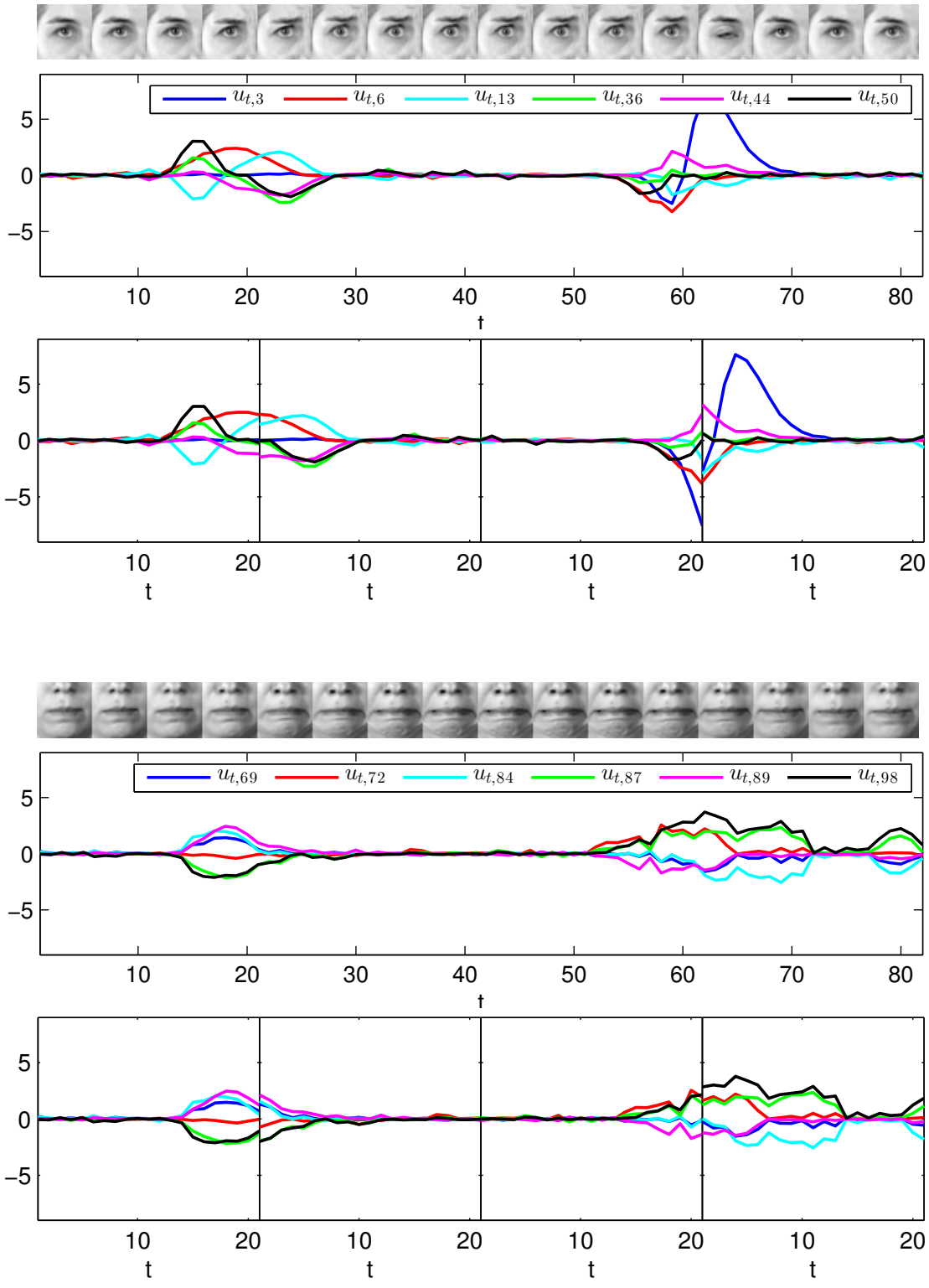


Figure B.3: Subject 1, expression of fear.

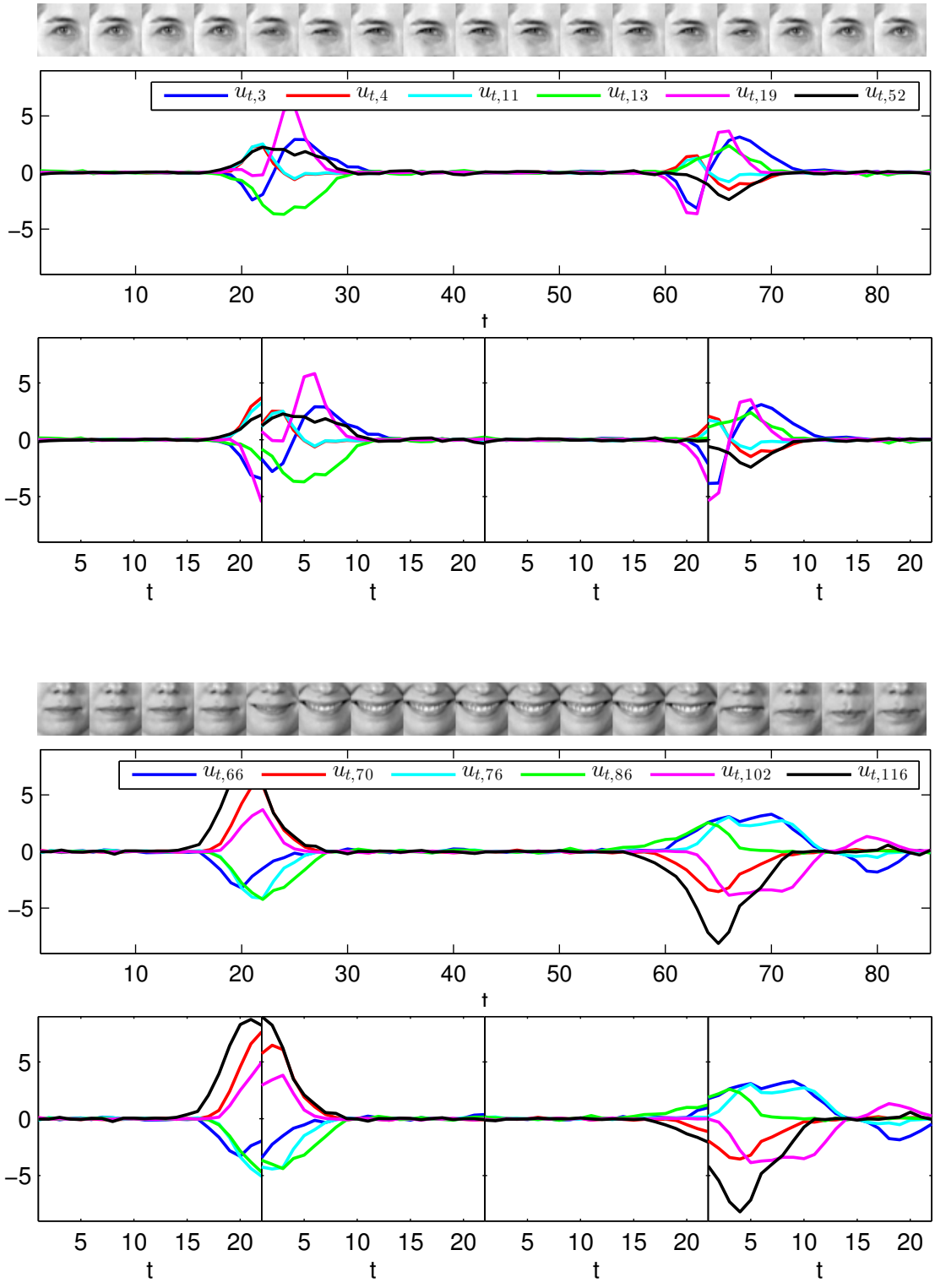


Figure B.4: Subject 1, expression of happiness.

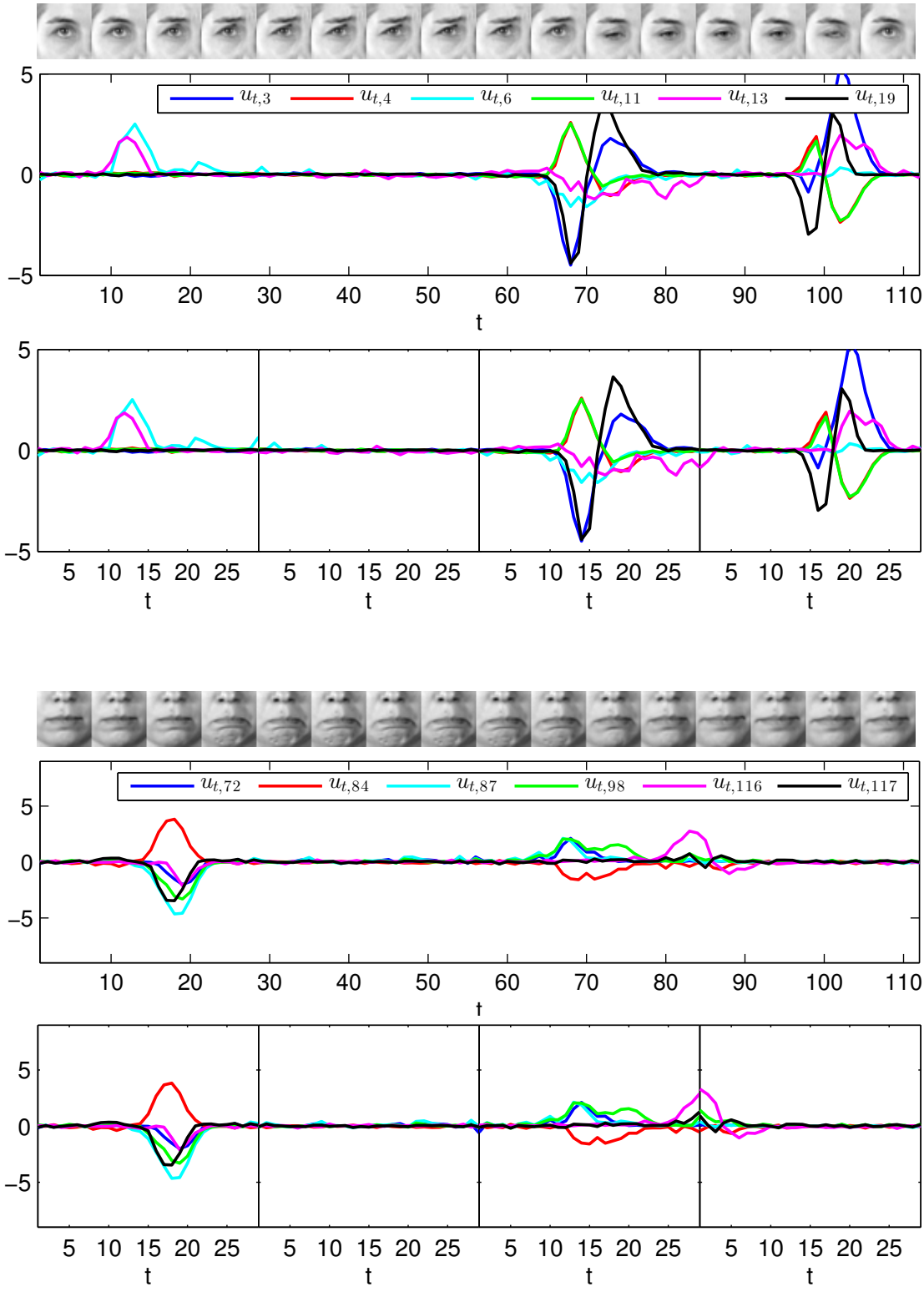


Figure B.5: Subject 1, expression of sadness.

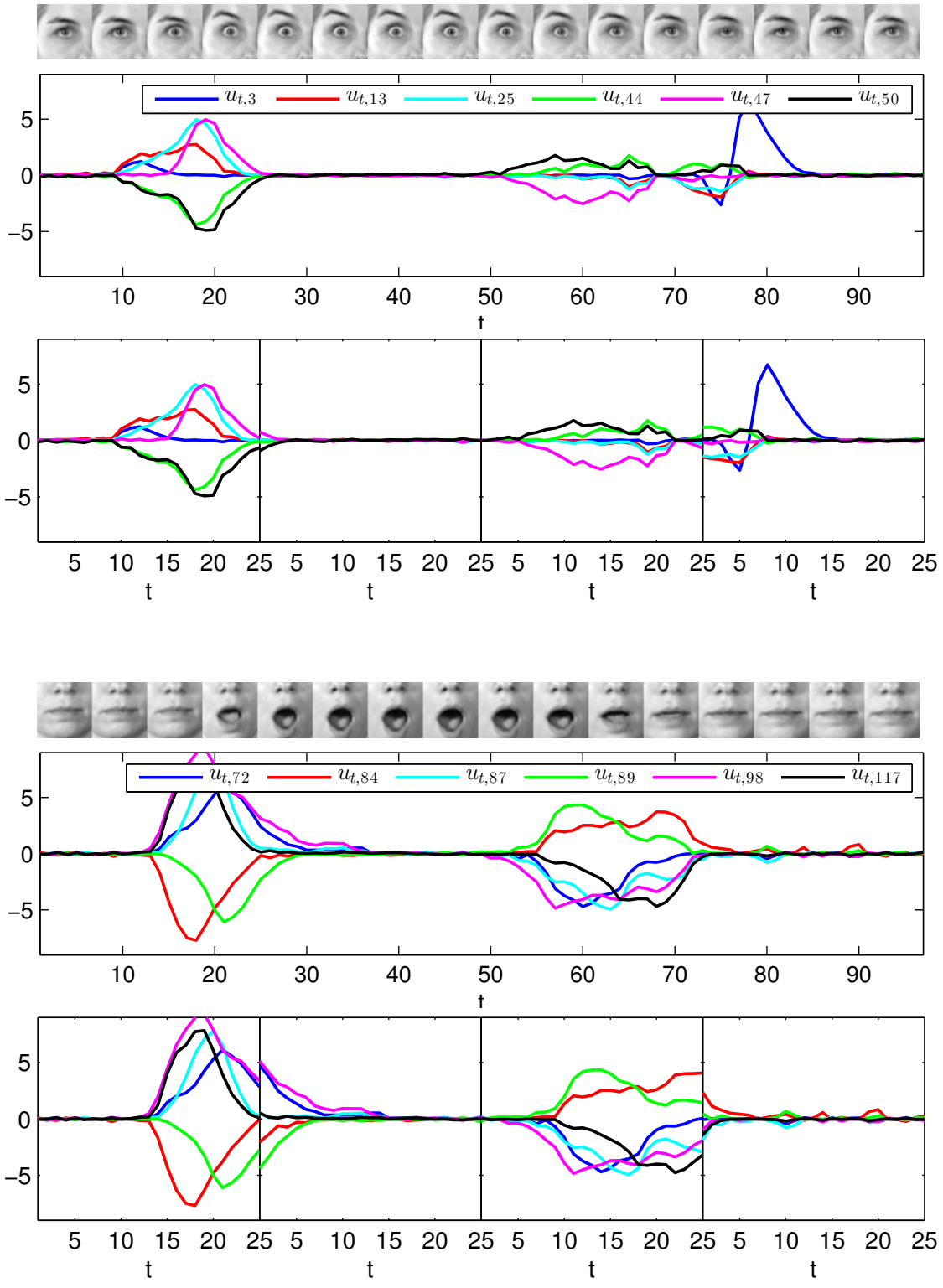


Figure B.6: Subject 1, expression of surprise.

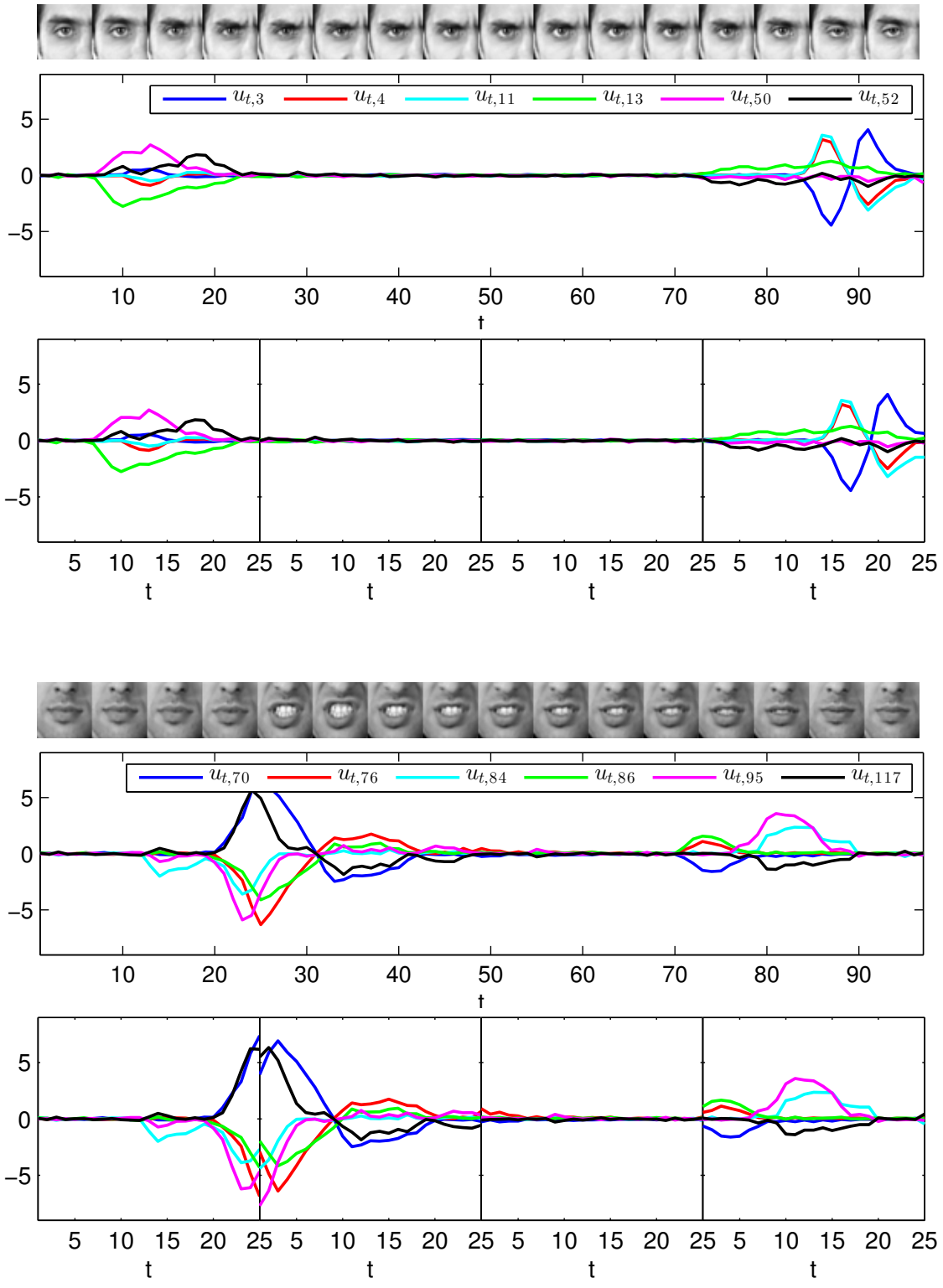


Figure B.7: Subject 2, expression of anger.

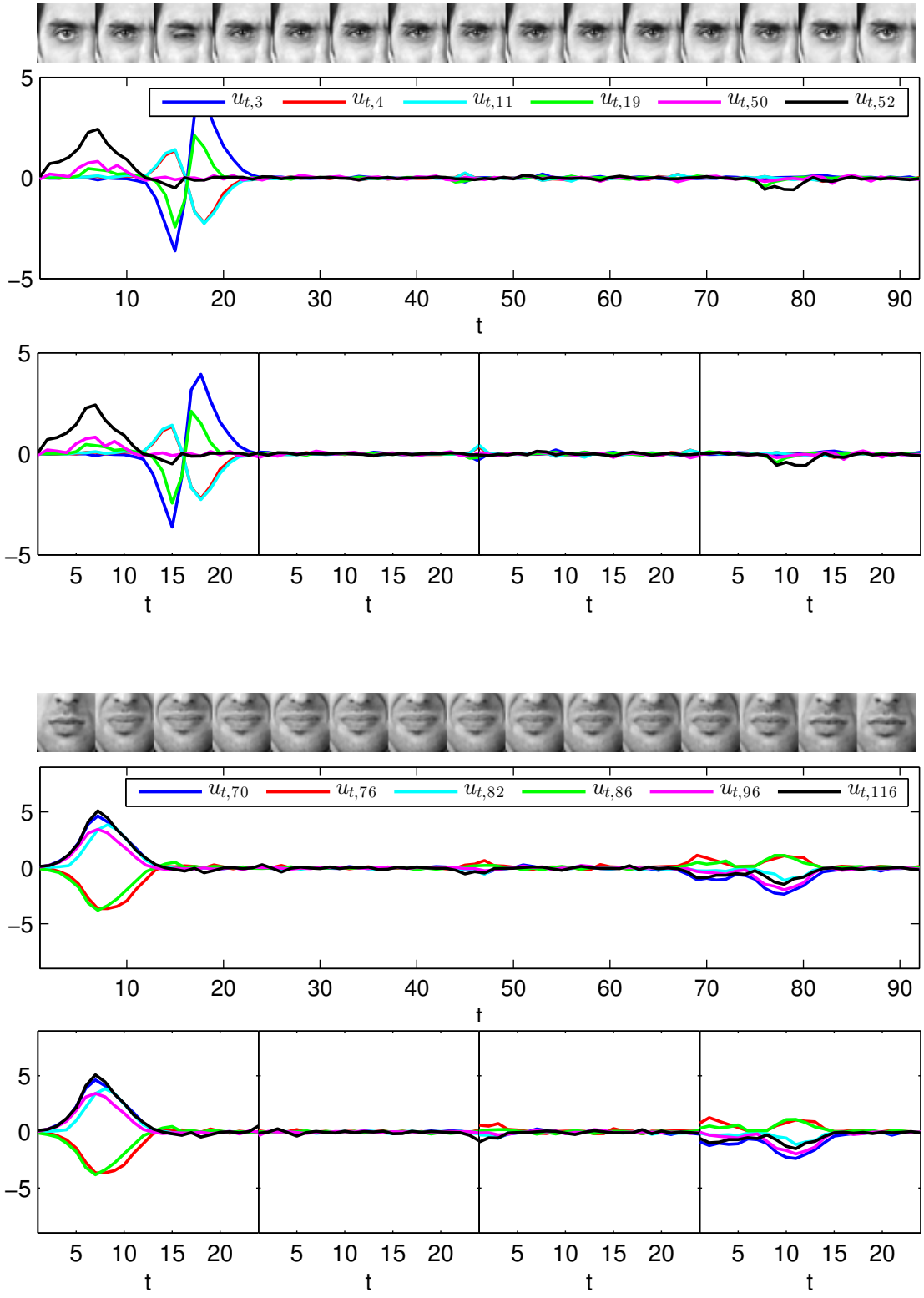


Figure B.8: Subject 2, expression of disgust.

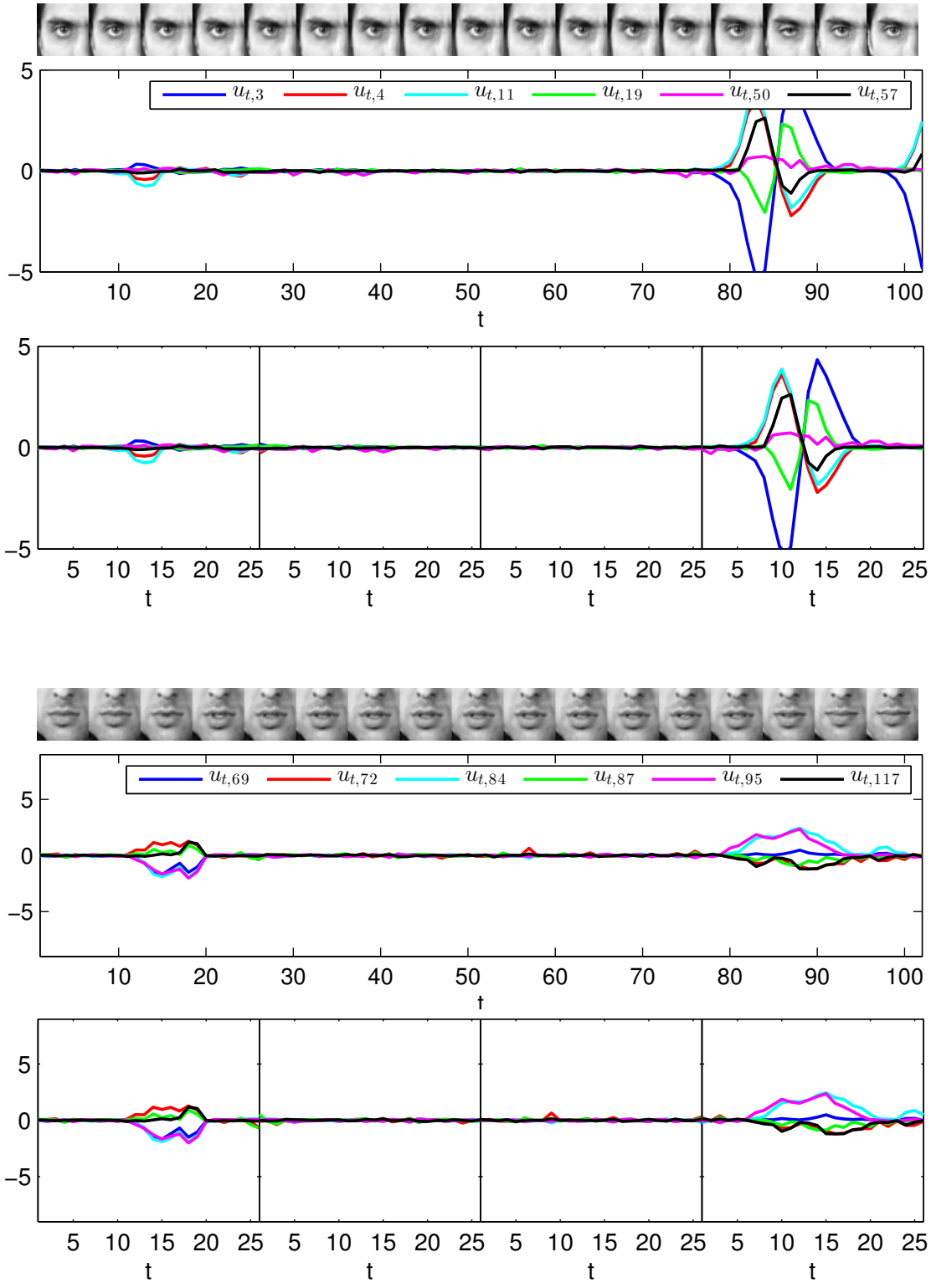


Figure B.9: Subject 2, expression of fear.

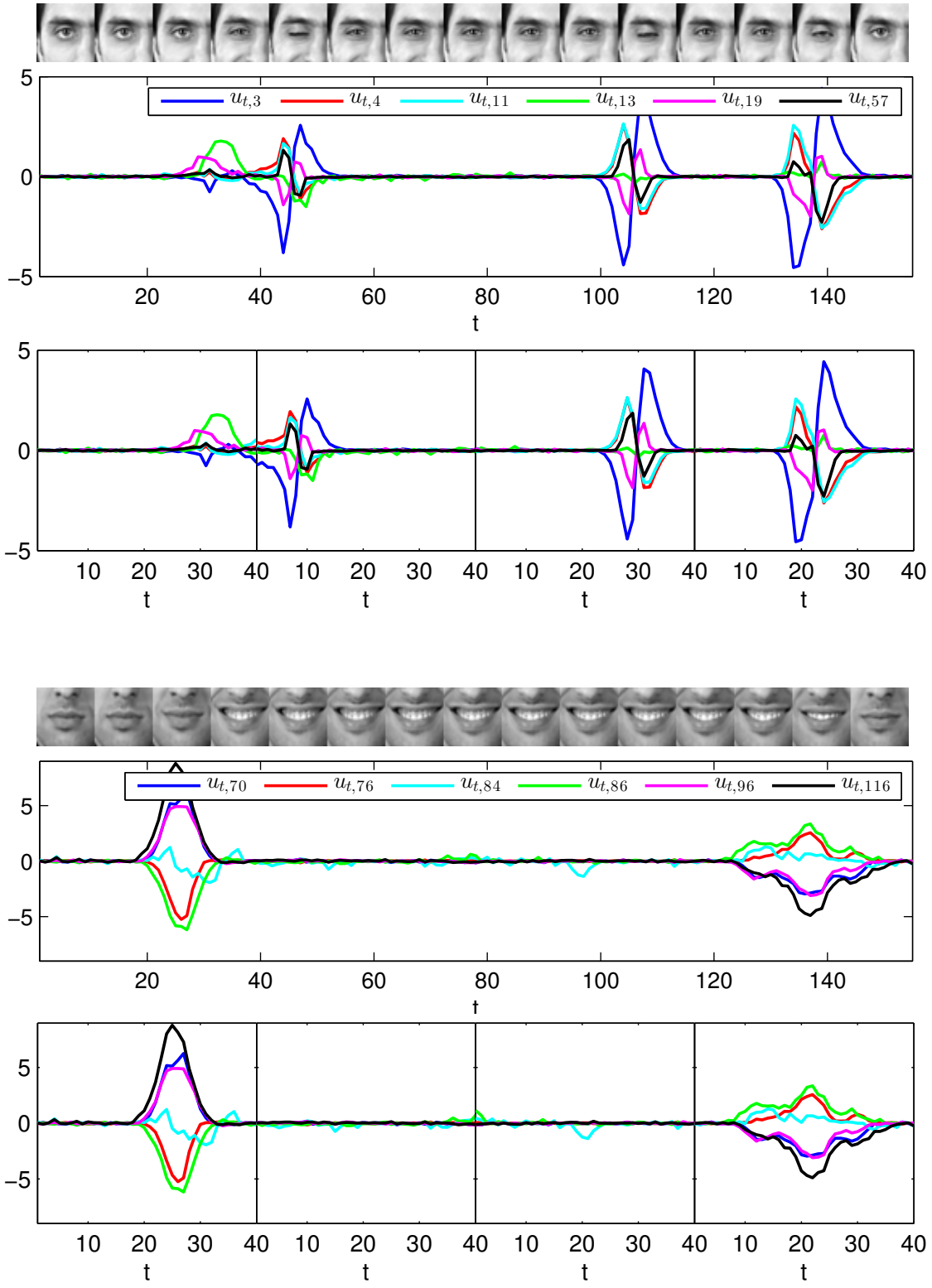


Figure B.10: Subject 2, expression of happiness.

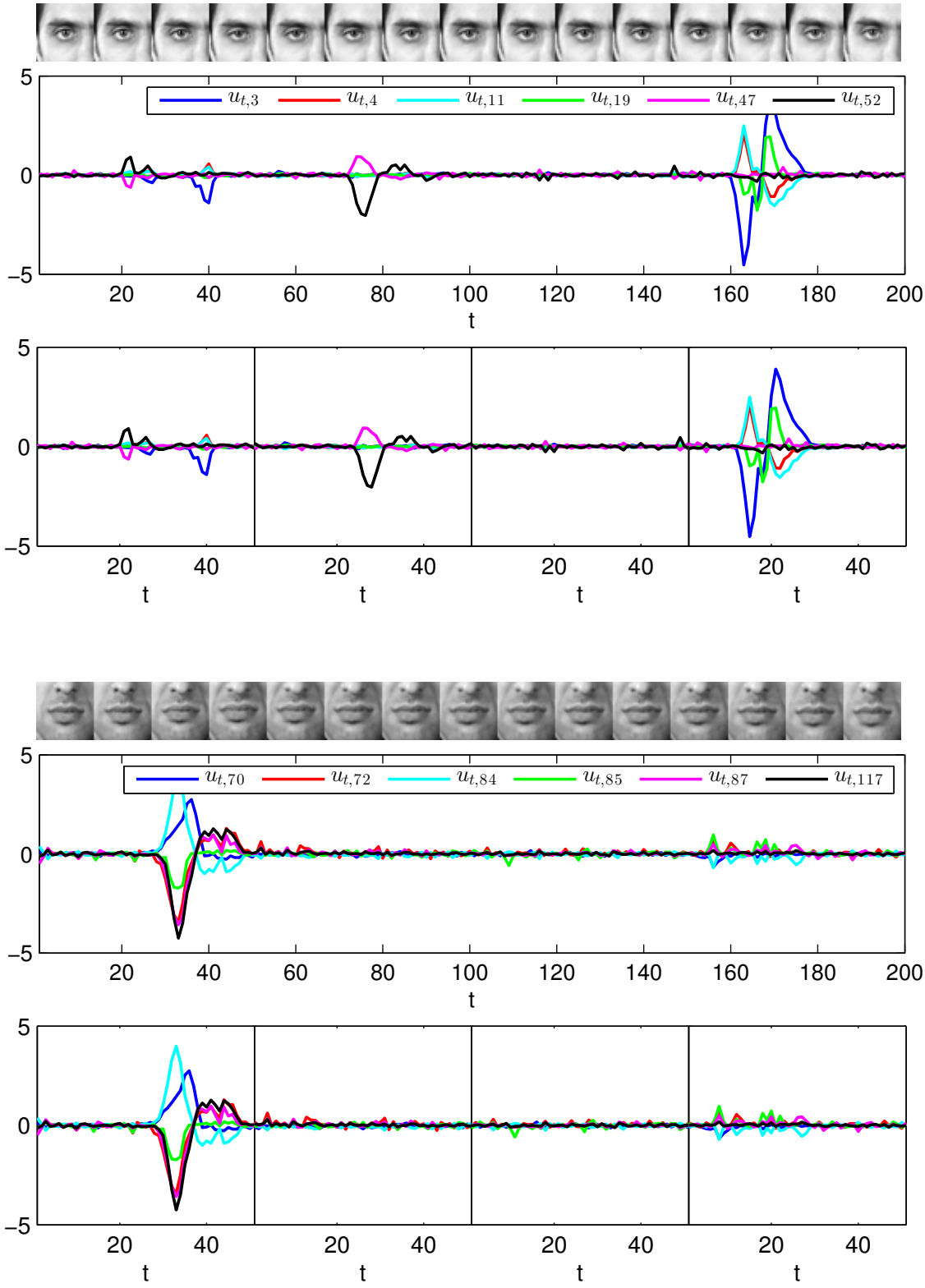


Figure B.11: Subject 2, expression of sadness.

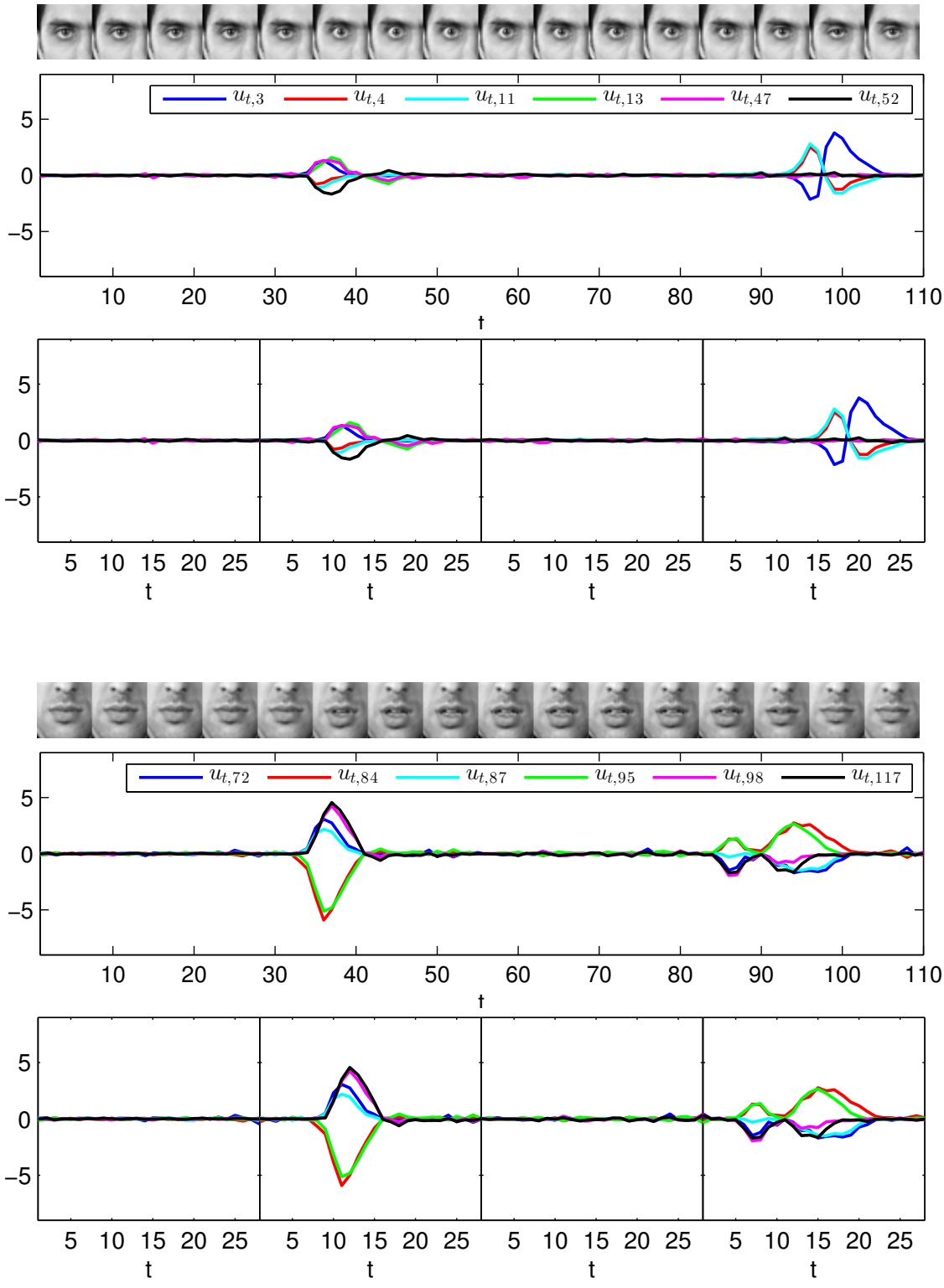


Figure B.12: Subject 2, expression of surprise.

Appendix C

Mathematica® Files

C.1 Application of Convolution Theorem for computing motion energy

This section provides the Mathematica® notebook file that is used to compute the Gabor motion energy of a moving line through the application of the Convolution theorem.

```
Clear[g, ge, go, f, x, y, t];(*filters and line*)
Clear[γ, λ, τ, σ, c];
(*fixed filter parameters-- Subscript[μ,t]=0 for good!*)
Clear[υg, υl, θg, θl]; (*speed of the filter (υg) and the line (υl),
orientation of the filter (θg) and the line (θl)*)(*The'source filter'*)
g[x_, y_, t_] = γ / (2 π σ^2) Exp[-((x^2 + y^2 + t^2) / (2 σ^2))]
  Cos[(2 π) / λ (x Cos[θg] + y Sin[θg] + υg t) + ϕ] 1 / (Sqrt[2 π] τ);

(*fix unrelated parameters*)
γ = 1; λ = 2 π; τ = σ = 1 / Sqrt[2]; c = 1;

(*(e)ven-x and (o)dd-phased filters for t=0 and t=1*)

assumptions = Im[u] == 0 && Im[v] == 0 && Im[z] == 0 && Im[x] == 0 &&
    Im[y] == 0 && Im[t] == 0 && Im[υl] == 0 && Im[υg] == 0 && Im[θg] == 0 &&
    Im[θl] == 0 && υg > 0 && υl > 0 && π > θg > 0 && π > θl > 0 && Im[Csc[θl]] == 0 &&
    Sin[θl] ≠ 0 && Cos[θl] ≠ 0 && Im[Cot[θl]] == 0 && ∞ > Cot[θl] > -∞ &&
    α ∈ Reals && β ∈ Reals && α > 0 && Im[cc] == 0 && Im[dd] == 0 && Cos[θl] > 0;

ge[x_, y_, t_] = g[x, y, t] /. {ϕ → 0}
(*go[x_,y_,t_]=g[x,y,t]/.{ϕ→π/2}*)

(*the line'generator'*)

f[x_, y_, t_] = c DiracDelta[x Cos[θl] - y Sin[θl] - υl t];

(* Fourier transforms of the filter and line respectively *)
fg = FourierTransform[ge[x, y, t], {x, y, t}, {w1, w2, w3}] // FullSimplify
ff = FourierTransform[f[x, y, t],
  {x, y, t}, {w1, w2, w3}, Assumptions → assumptions]

(* Multiply line with filter in the Fourier domain *)
fr = ff * fg // FullSimplify
(* THIS IS NOT USED DIRECTLY! It is restructured manually,
specifically, the Cosh[]∓Sinh[]'s are converted to Exp[∓]'s
     SEE NEXT CELL! *)
```

$$\frac{e^{-t^2-x^2-y^2}\,\text{Cos}[t\,\upsilon g + x\,\text{Cos}[\theta g] + y\,\text{Sin}[\theta g]]}{\pi^{3/2}}$$

$$\frac{1}{4\sqrt{2}\,\pi^{3/2}}\,(1+\text{Cosh}[w3\,\upsilon g + w1\,\text{Cos}[\theta g] + w2\,\text{Sin}[\theta g]] + \text{Sinh}[w3\,\upsilon g + w1\,\text{Cos}[\theta g] + w2\,\text{Sin}[\theta g]])$$
$$\left(\text{Cosh}\left[\frac{1}{4}\left(1 + w1^2 + w2^2 + (w3+\upsilon g)^2 + 2\,w1\,\text{Cos}[\theta g] + 2\,w2\,\text{Sin}[\theta g]\right)\right] - \right.$$
$$\left.\text{Sinh}\left[\frac{1}{4}\left(1 + w1^2 + w2^2 + (w3+\upsilon g)^2 + 2\,w1\,\text{Cos}[\theta g] + 2\,w2\,\text{Sin}[\theta g]\right)\right]\right)$$

$$\frac{1}{4\pi}\,\text{DiracDelta}[w3 + w1\,\upsilon l\,\text{Sec}[\theta l]]\,\text{DiracDelta}[w2 + w1\,\text{Tan}[\theta l]]\,\text{Sec}[\theta l]$$
$$(1 + \text{Cosh}[w3\,\upsilon g + w1\,\text{Cos}[\theta g] + w2\,\text{Sin}[\theta g]] + \text{Sinh}[w3\,\upsilon g + w1\,\text{Cos}[\theta g] + w2\,\text{Sin}[\theta g]])$$
$$\left(\text{Cosh}\left[\frac{1}{4}\left(1 + w1^2 + w2^2 + (w3 + \upsilon g)^2 + 2\,w1\,\text{Cos}[\theta g] + 2\,w2\,\text{Sin}[\theta g]\right)\right] - \right.$$
$$\left.\text{Sinh}\left[\frac{1}{4}\left(1 + w1^2 + w2^2 + (w3 + \upsilon g)^2 + 2\,w1\,\text{Cos}[\theta g] + 2\,w2\,\text{Sin}[\theta g]\right)\right]\right)$$

```
fgRes = 1/(4 Sqrt[2] π^(3/2)) (1 + Exp[w3 υg + w1 Cos[θg] + w2 Sin[θg]])
  Exp[-1/4 (1 + w1^2 + w2^2 + (w3 + υg)^2 + 2 w1 Cos[θg] + 2 w2 Sin[θg])]
```

$$\frac{e^{\frac{1}{4}\left(-1-w1^2-w2^2-(w3+\upsilon g)^2-2\,w1\,\text{Cos}[\theta g]-2\,w2\,\text{Sin}[\theta g]\right)}\left(1 + e^{w3\,\upsilon g+w1\,\text{Cos}[\theta g]+w2\,\text{Sin}[\theta g]}\right)}{4\sqrt{2}\,\pi^{3/2}}$$

```
(* Restructure fourier output,
and then compute the inverse -- which is the output of convolution *)
resFr = 1/(4 π) DiracDelta[w3 + w1 υl Sec[θl]]
  DiracDelta[w2 + w1 Tan[θl]] Sec[θl] (1 + Exp[w3 υg + w1 Cos[θg] + w2 Sin[θg]])
  (Exp[-1/4 (1 + w1^2 + w2^2 + (w3 + υg)^2 + 2 w1 Cos[θg] + 2 w2 Sin[θg])])
ifr = InverseFourierTransform[resFr, {w2, w1, w3}, {y, x, t}];
```

$$\frac{1}{4\pi}\,e^{\frac{1}{4}\left(-1-w1^2-w2^2-(w3+\upsilon g)^2-2\,w1\,\text{Cos}[\theta g]-2\,w2\,\text{Sin}[\theta g]\right)}\left(1 + e^{w3\,\upsilon g+w1\,\text{Cos}[\theta g]+w2\,\text{Sin}[\theta g]}\right)$$
$$\text{DiracDelta}[w3 + w1\,\upsilon l\,\text{Sec}[\theta l]]\,\text{DiracDelta}[w2 + w1\,\text{Tan}[\theta l]]\,\text{Sec}[\theta l]$$

```
(* ifr is a very long expression,
needs simplification. Simplify[] here takes very long (hours) but works. *)
sifr = Simplify[ifr, Assumptions → assumptions, TimeConstraint → 300 000]
```

$$\frac{1}{8\sqrt{2}\,\pi^2\sqrt{1 + \frac{1}{\upsilon l^2}\,\upsilon l\,\text{Abs}[\text{Sec}[\theta l]]}}\,\text{Sec}[\theta l]\left(\left(\left(\sqrt{1 + \frac{1}{\upsilon l^2}}\,\upsilon l\right.\right.\right.$$
$$\left.\left.\left.\text{Erf}\left[\frac{1}{2}\sqrt{\frac{(2\,i\,t\,\upsilon l - \upsilon g\,\upsilon l - 2\,i\,x\,\text{Cos}[\theta l] + \text{Cos}[\theta g + \theta l] + 2\,i\,y\,\text{Sin}[\theta l])^2}{1 + \upsilon l^2}}\right]\right.\right.$$
$$(2\,i\,t\,\upsilon l - \upsilon g\,\upsilon l - 2\,i\,x\,\text{Cos}[\theta l] + \text{Cos}[\theta g + \theta l] + 2\,i\,y\,\text{Sin}[\theta l]) +$$
$$\left.\left(2 - i\,\text{Erfi}\left[\frac{2\,t\,\upsilon l + i\,\upsilon g\,\upsilon l - 2\,x\,\text{Cos}[\theta l] - i\,\text{Cos}[\theta g + \theta l] + 2\,y\,\text{Sin}[\theta l]}{2\sqrt{1 + \upsilon l^2}}\right]\right)\right.$$
$$\left.\sqrt{-\left(1 + \upsilon l^2\right)(2\,t\,\upsilon l + i\,\upsilon g\,\upsilon l - 2\,x\,\text{Cos}[\theta l] - i\,\text{Cos}[\theta g + \theta l] + 2\,y\,\text{Sin}[\theta l])^2}\right)$$
$$\left(\text{Cosh}\left[1 \Big/ \left(8\left(1 + \upsilon l^2\right)\right)\right]\left(1 + 4\,x^2 + 4\,y^2 + 2\,\upsilon g^2 + 2\,\upsilon l^2 + 8\,t^2\,\upsilon l^2 + 8\,i\,t\,\upsilon g\,\upsilon l^2 + \right.\right.$$
$$\left.\left.4\,i\,x\,\text{Cos}[\theta g] - 8\,x\,(2\,t + i\,\upsilon g)\,\upsilon l\,\text{Cos}[\theta l] + 4\,x^2\,\text{Cos}[2\,\theta l] - \right.\right.$$

$$
\begin{aligned}
&4\,y^2\,\text{Cos}[2\,\theta l] - 8\,i\,t\,\upsilon l\,\text{Cos}[\theta g+\theta l] + 4\,\upsilon g\,\upsilon l\,\text{Cos}[\theta g+\theta l] -\\
&\text{Cos}[2\,(\theta g+\theta l)] + 4\,i\,x\,\text{Cos}[\theta g+2\,\theta l] + 4\,i\,y\,\text{Sin}[\theta g] + 16\,t\,y\,\upsilon l\,\text{Sin}[\theta l] +\\
&8\,i\,y\,\upsilon g\,\upsilon l\,\text{Sin}[\theta l] - 8\,x\,y\,\text{Sin}[2\,\theta l] - 4\,i\,y\,\text{Sin}[\theta g+2\,\theta l]\Big)\Big] -\\
&\text{Sinh}\Big[1\Big/\Big(8\,\big(1+\upsilon l^2\big)\Big)\Big(1+4\,x^2+4\,y^2+2\,\upsilon g^2+2\,\upsilon l^2+8\,t^2\,\upsilon l^2+8\,i\,t\,\upsilon g\,\upsilon l^2+\\
&4\,i\,x\,\text{Cos}[\theta g] - 8\,x\,(2\,t+i\,\upsilon g)\,\upsilon l\,\text{Cos}[\theta l] + 4\,x^2\,\text{Cos}[2\,\theta l] -\\
&4\,y^2\,\text{Cos}[2\,\theta l] - 8\,i\,t\,\upsilon l\,\text{Cos}[\theta g+\theta l] + 4\,\upsilon g\,\upsilon l\,\text{Cos}[\theta g+\theta l] -\\
&\text{Cos}[2\,(\theta g+\theta l)] + 4\,i\,x\,\text{Cos}[\theta g+2\,\theta l] + 4\,i\,y\,\text{Sin}[\theta g] + 16\,t\,y\,\upsilon l\,\text{Sin}[\theta l] +\\
&8\,i\,y\,\upsilon g\,\upsilon l\,\text{Sin}[\theta l] - 8\,x\,y\,\text{Sin}[2\,\theta l] - 4\,i\,y\,\text{Sin}[\theta g+2\,\theta l]\Big)\Big]\Big)\Bigg)\Bigg/\\
&\Big(\sqrt{-\big(1+\upsilon l^2\big)\,(2\,t\,\upsilon l+i\,\upsilon g\,\upsilon l-2\,x\,\text{Cos}[\theta l]-i\,\text{Cos}[\theta g+\theta l]+2\,y\,\text{Sin}[\theta l])^2}\Big) +\\
&\Bigg(\Bigg(\Bigg(\text{Erf}\Big[\frac{1}{2}\sqrt{\frac{(2\,i\,t\,\upsilon l+\upsilon g\,\upsilon l-2\,i\,x\,\text{Cos}[\theta l]-\text{Cos}[\theta g+\theta l]+2\,i\,y\,\text{Sin}[\theta l])^2}{1+\upsilon l^2}}\Big]\\
&\sqrt{(2\,i\,t\,\upsilon l+\upsilon g\,\upsilon l-2\,i\,x\,\text{Cos}[\theta l]-\text{Cos}[\theta g+\theta l]+2\,i\,y\,\text{Sin}[\theta l])^2} +\\
&\Big(2\,i+\text{Erfi}\Big[\frac{2\,t\,\upsilon l-i\,\upsilon g\,\upsilon l-2\,x\,\text{Cos}[\theta l]+i\,\text{Cos}[\theta g+\theta l]+2\,y\,\text{Sin}[\theta l]}{2\sqrt{1+\upsilon l^2}}\Big]\Big)\\
&(2\,t\,\upsilon l-i\,\upsilon g\,\upsilon l-2\,x\,\text{Cos}[\theta l]+i\,\text{Cos}[\theta g+\theta l]+2\,y\,\text{Sin}[\theta l])\Bigg)\\
&\Big(\text{Cosh}\Big[1\Big/\Big(8\,\big(1+\upsilon l^2\big)\Big)\Big(1+4\,x^2+4\,y^2+2\,\upsilon g^2+2\,\upsilon l^2+8\,t^2\,\upsilon l^2-8\,i\,t\,\upsilon g\,\upsilon l^2-\\
&4\,i\,x\,\text{Cos}[\theta g] - 8\,x\,(2\,t-i\,\upsilon g)\,\upsilon l\,\text{Cos}[\theta l] + 4\,x^2\,\text{Cos}[2\,\theta l] -\\
&4\,y^2\,\text{Cos}[2\,\theta l] + 8\,i\,t\,\upsilon l\,\text{Cos}[\theta g+\theta l] + 4\,\upsilon g\,\upsilon l\,\text{Cos}[\theta g+\theta l] -\\
&\text{Cos}[2\,(\theta g+\theta l)] - 4\,i\,x\,\text{Cos}[\theta g+2\,\theta l] - 4\,i\,y\,\text{Sin}[\theta g] + 16\,t\,y\,\upsilon l\,\text{Sin}[\theta l] -\\
&8\,i\,y\,\upsilon g\,\upsilon l\,\text{Sin}[\theta l] - 8\,x\,y\,\text{Sin}[2\,\theta l] + 4\,i\,y\,\text{Sin}[\theta g+2\,\theta l]\Big)\Big] -\\
&\text{Sinh}\Big[1\Big/\Big(8\,\big(1+\upsilon l^2\big)\Big)\Big(1+4\,x^2+4\,y^2+2\,\upsilon g^2+2\,\upsilon l^2+8\,t^2\,\upsilon l^2-8\,i\,t\,\upsilon g\,\upsilon l^2-\\
&4\,i\,x\,\text{Cos}[\theta g] - 8\,x\,(2\,t-i\,\upsilon g)\,\upsilon l\,\text{Cos}[\theta l] + 4\,x^2\,\text{Cos}[2\,\theta l] -\\
&4\,y^2\,\text{Cos}[2\,\theta l] + 8\,i\,t\,\upsilon l\,\text{Cos}[\theta g+\theta l] + 4\,\upsilon g\,\upsilon l\,\text{Cos}[\theta g+\theta l] -\\
&\text{Cos}[2\,(\theta g+\theta l)] - 4\,i\,x\,\text{Cos}[\theta g+2\,\theta l] - 4\,i\,y\,\text{Sin}[\theta g] + 16\,t\,y\,\upsilon l\,\text{Sin}[\theta l] -\\
&8\,i\,y\,\upsilon g\,\upsilon l\,\text{Sin}[\theta l] - 8\,x\,y\,\text{Sin}[2\,\theta l] + 4\,i\,y\,\text{Sin}[\theta g+2\,\theta l]\Big)\Big]\Big)\Bigg)\Bigg/\\
&(2\,i\,t\,\upsilon l+\upsilon g\,\upsilon l-2\,i\,x\,\text{Cos}[\theta l]-\text{Cos}[\theta g+\theta l]+2\,i\,y\,\text{Sin}[\theta l])\Bigg)
\end{aligned}
$$

```
(* The output of Simplify is still very long,
needs further simplification (done manually).
   Simplify_3DConvolution.nb is written for this purpose. *)
```

C.2 Simplification of Fourier Transform output for completing application of Convolution Theorem

The outcome of the Convolution Theorem in Section C.1 is cluttered and Mathematica® cannot simplify it further fully automatically. We have completed the simplification partly through manual computation. This section provides the Mathematica® notebook file that leads to the simplified version of the convolution output presented in Section C.1.

```
(* This script defines a procedure
  to simplify the output of the 3D convolution
   of a moving line with a spatio-temporal Gabor filter *)

(* This cell defines some functions for dealing with the simplification,
   mainly about reformulating exponential expressions *)
ClearAll["Global`*"]

(* Multiply all elements in a list *)
listProduct[x_List] := Times @@ x

(* Check if an expression contains i *)
ImaginaryQ[expr_] := ! FreeQ[expr, _Complex]

(* restructure the exponential term to keep ONLY IMAGINARY PARTS!
    The argument of the exponential must have been Expand[]'ed IN ADVANCE!
    i.e. required form Exp[a/2+b/2] rather than e.g. Exp[(a+b)/2]

    First obtain a list only of all imaginary
    components and then multiply all the elements in the list
    ##&[]: This is the expression that destroys itself
  (and is removed automatically from the list) *)
restExpImag[x_] := Table[ If[ ImaginaryQ[ x[[2]][[n]]], Exp[x[[2]][[n]]] //
     ExpToTrig, ## &[]], {n, 1, Length[x[[All]][[2]]]}] // listProduct

(* restructure the exponential term to keep ONLY REAL PARTS!
   The argument of the exponential must have been Expand[]'ed IN ADVANCE!
   i.e. required form Exp[a/2+b/2] rather than e.g. Exp[(a+b)/2] *)
restExpReal[x_] := Table[ If[ ImaginaryQ[ x[[2]][[n]]], ## &[], Exp[x[[2]][[n]]]],
   {n, 1, Length[x[[All]][[2]]]}] // listProduct

(* Simplify an exponential expression and keep using the routines above!
   This outputs a form where complex coeffs of the exp. are a
   converted to trig. expressions while real coeffs are kept exp.
   e.g. output: (Cos[a]-i Sin[b])Exp[c]
    PS: Something is probably wrong in the functions below,
they are not used in the script anyway.
          But they must be checked if they are going to be used!
*)
simplifyExpImag[x_] := Module[{expr, arg},
  expr = Simplify[restExpImag[x]];
  arg = Last[First[expr]] // FullSimplify;
  Cos[arg] + First[Last[expr]] Sin[arg]
 ]

(* Sometimes the above cannot handle complicated expressions
   and the output contains Cosh[] Sinh[] functions (which were supposed
   to be raised in the exp. expression) *)
fullSimplifyExpImag[x_] := Module[{expr, arg},
  expr = FullSimplify[restExpImag[x]];
  arg = Last[First[expr]] // FullSimplify;
  Cos[arg] + First[Last[expr]] Sin[arg]
```

```
 ]

(* Use the functions above to fully  *)
restExpFullSimplified[x_] := fullSimplifyExpImag[x] restExpReal[x]
restExpSimplified[x_] := simplifyExpImag[x] restExpReal[x]

(* The expression to simplify. This is the output
 of the 3D convolution of a moving line with a Gabor
 filter (obtained from the file convolution3D_fourier.nb).
   It is obtained by using the convolution theorem. The expression below
 is the output of the InverseFourierTransform (Simplify[]'ed once).  *)
e = (1 / (8 Sqrt[2] π^2 Sqrt[1 + 1 / υl^2] υl Abs[Sec[θl]])) Sec[θl]
  (((Sqrt[1 + 1 / υl^2] υl Erf[1 / 2 Sqrt[(2 I t υl - υg υl - 2 I x Cos[θl] +
                    Cos[θg + θl] + 2 I y Sin[θl])^2 / (1 + υl^2)]]
           (2 I t υl - υg υl - 2 I x Cos[θl] + Cos[θg + θl] + 2 I y Sin[θl]) +
         (2 - I Erfi[(2 t υl + I υg υl - 2 x Cos[θl] - I Cos[θg + θl] + 2 y Sin[θl]) /
               (2 Sqrt[1 + υl^2])]) Sqrt[-(1 + υl^2)
             (2 t υl + I υg υl - 2 x Cos[θl] - I Cos[θg + θl] + 2 y Sin[θl])^2])
       (Cosh[1 / (8 (1 + υl^2)) (1 + 4 x^2 + 4 y^2 + 2 υg^2 + 2 υl^2 + 8 t^2 υl^2 +
             8 I t υg υl^2 + 4 I x Cos[θg] - 8 x (2 t + I υg) υl Cos[θl] + 4 x^2 Cos[2 θl] -
             4 y^2 Cos[2 θl] - 8 I t υl Cos[θg + θl] + 4 υg υl Cos[θg + θl] -
             Cos[2 (θg + θl)] + 4 I x Cos[θg + 2 θl] + 4 I y Sin[θg] + 16 t y υl Sin[θl] +
             8 I y υg υl Sin[θl] - 8 x y Sin[2 θl] - 4 I y Sin[θg + 2 θl])] -
         Sinh[1 / (8 (1 + υl^2)) (1 + 4 x^2 + 4 y^2 + 2 υg^2 + 2 υl^2 + 8 t^2 υl^2 +
             8 I t υg υl^2 + 4 I x Cos[θg] - 8 x (2 t + I υg) υl Cos[θl] + 4 x^2 Cos[2 θl] -
             4 y^2 Cos[2 θl] - 8 I t υl Cos[θg + θl] + 4 υg υl Cos[θg + θl] -
             Cos[2 (θg + θl)] + 4 I x Cos[θg + 2 θl] + 4 I y Sin[θg] + 16 t y υl Sin[θl] +
             8 I y υg υl Sin[θl] - 8 x y Sin[2 θl] - 4 I y Sin[θg + 2 θl])])) /
     (Sqrt[-(1 + υl^2) (2 t υl + I υg υl - 2 x Cos[θl] - I Cos[θg + θl] + 2 y Sin[θl])^2]) +
    ((Erf[1 / 2 Sqrt[(2 I t υl + υg υl - 2 I x Cos[θl] -
                    Cos[θg + θl] + 2 I y Sin[θl])^2 / (1 + υl^2)]]
          Sqrt[(2 I t υl + υg υl - 2 I x Cos[θl] - Cos[θg + θl] + 2 I y Sin[θl])^2] +
         (2 I + Erfi[(2 t υl - I υg υl - 2 x Cos[θl] + I Cos[θg + θl] +
               2 y Sin[θl]) / (2 Sqrt[1 + υl^2])])
          (2 t υl - I υg υl - 2 x Cos[θl] + I Cos[θg + θl] + 2 y Sin[θl]))
       (Cosh[1 / (8 (1 + υl^2)) (1 + 4 x^2 + 4 y^2 + 2 υg^2 + 2 υl^2 + 8 t^2 υl^2 -
             8 I t υg υl^2 - 4 I x Cos[θg] - 8 x (2 t - I υg) υl Cos[θl] + 4 x^2 Cos[2 θl] -
             4 y^2 Cos[2 θl] + 8 I t υl Cos[θg + θl] + 4 υg υl Cos[θg + θl] -
             Cos[2 (θg + θl)] - 4 I x Cos[θg + 2 θl] - 4 I y Sin[θg] + 16 t y υl Sin[θl] -
             8 I y υg υl Sin[θl] - 8 x y Sin[2 θl] + 4 I y Sin[θg + 2 θl])] -
         Sinh[1 / (8 (1 + υl^2)) (1 + 4 x^2 + 4 y^2 + 2 υg^2 + 2 υl^2 + 8 t^2 υl^2 -
             8 I t υg υl^2 - 4 I x Cos[θg] - 8 x (2 t - I υg) υl Cos[θl] + 4 x^2 Cos[2 θl] -
             4 y^2 Cos[2 θl] + 8 I t υl Cos[θg + θl] + 4 υg υl Cos[θg + θl] -
             Cos[2 (θg + θl)] - 4 I x Cos[θg + 2 θl] - 4 I y Sin[θg] + 16 t y υl Sin[θl] -
             8 I y υg υl Sin[θl] - 8 x y Sin[2 θl] + 4 I y Sin[θg + 2 θl])])) /
     (2 I t υl + υg υl - 2 I x Cos[θl] - Cos[θg + θl] + 2 I y Sin[θl]))
```

$$\frac{1}{8\sqrt{2}\,\pi^2\sqrt{1+\frac{1}{\upsilon l^2}}\,\upsilon l\,\text{Abs}[\text{Sec}[\theta l]]}\,\text{Sec}[\theta l]\left(\left(\left(\left(\sqrt{1+\frac{1}{\upsilon l^2}}\,\upsilon l\right.\right.\right.\right.$$

$$\mathrm{Erf}\left[\frac{1}{2}\sqrt{\frac{(2\,i\,t\,\upsilon l-\upsilon g\,\upsilon l-2\,i\,x\,\mathrm{Cos}[\theta l]+\mathrm{Cos}[\theta g+\theta l]+2\,i\,y\,\mathrm{Sin}[\theta l])^2}{1+\upsilon l^2}}\right]$$
$$(2\,i\,t\,\upsilon l-\upsilon g\,\upsilon l-2\,i\,x\,\mathrm{Cos}[\theta l]+\mathrm{Cos}[\theta g+\theta l]+2\,i\,y\,\mathrm{Sin}[\theta l])+$$
$$\left(2-i\,\mathrm{Erfi}\left[\frac{2\,t\,\upsilon l+i\,\upsilon g\,\upsilon l-2\,x\,\mathrm{Cos}[\theta l]-i\,\mathrm{Cos}[\theta g+\theta l]+2\,y\,\mathrm{Sin}[\theta l]}{2\sqrt{1+\upsilon l^2}}\right]\right)$$
$$\sqrt{\left(-1-\upsilon l^2\right)(2\,t\,\upsilon l+i\,\upsilon g\,\upsilon l-2\,x\,\mathrm{Cos}[\theta l]-i\,\mathrm{Cos}[\theta g+\theta l]+2\,y\,\mathrm{Sin}[\theta l])^2}\Bigg)$$
$$\Big(\mathrm{Cosh}\Big[1\Big/\left(8\left(1+\upsilon l^2\right)\right)\left(1+4\,x^2+4\,y^2+2\,\upsilon g^2+2\,\upsilon l^2+8\,t^2\,\upsilon l^2+8\,i\,t\,\upsilon g\,\upsilon l^2+\right.$$
$$4\,i\,x\,\mathrm{Cos}[\theta g]-8\,x\,(2\,t+i\,\upsilon g)\,\upsilon l\,\mathrm{Cos}[\theta l]+4\,x^2\,\mathrm{Cos}[2\,\theta l]-$$
$$4\,y^2\,\mathrm{Cos}[2\,\theta l]-8\,i\,t\,\upsilon l\,\mathrm{Cos}[\theta g+\theta l]+4\,\upsilon g\,\upsilon l\,\mathrm{Cos}[\theta g+\theta l]-$$
$$\mathrm{Cos}[2\,(\theta g+\theta l)]+4\,i\,x\,\mathrm{Cos}[\theta g+2\,\theta l]+4\,i\,y\,\mathrm{Sin}[\theta g]+16\,t\,y\,\upsilon l\,\mathrm{Sin}[\theta l]+$$
$$\left.8\,i\,y\,\upsilon g\,\upsilon l\,\mathrm{Sin}[\theta l]-8\,x\,y\,\mathrm{Sin}[2\,\theta l]-4\,i\,y\,\mathrm{Sin}[\theta g+2\,\theta l]\right)\Big]-$$
$$\mathrm{Sinh}\Big[1\Big/\left(8\left(1+\upsilon l^2\right)\right)\left(1+4\,x^2+4\,y^2+2\,\upsilon g^2+2\,\upsilon l^2+8\,t^2\,\upsilon l^2+8\,i\,t\,\upsilon g\,\upsilon l^2+\right.$$
$$4\,i\,x\,\mathrm{Cos}[\theta g]-8\,x\,(2\,t+i\,\upsilon g)\,\upsilon l\,\mathrm{Cos}[\theta l]+4\,x^2\,\mathrm{Cos}[2\,\theta l]-$$
$$4\,y^2\,\mathrm{Cos}[2\,\theta l]-8\,i\,t\,\upsilon l\,\mathrm{Cos}[\theta g+\theta l]+4\,\upsilon g\,\upsilon l\,\mathrm{Cos}[\theta g+\theta l]-$$
$$\mathrm{Cos}[2\,(\theta g+\theta l)]+4\,i\,x\,\mathrm{Cos}[\theta g+2\,\theta l]+4\,i\,y\,\mathrm{Sin}[\theta g]+16\,t\,y\,\upsilon l\,\mathrm{Sin}[\theta l]+$$
$$\left.8\,i\,y\,\upsilon g\,\upsilon l\,\mathrm{Sin}[\theta l]-8\,x\,y\,\mathrm{Sin}[2\,\theta l]-4\,i\,y\,\mathrm{Sin}[\theta g+2\,\theta l]\right)\Big]\Bigg)\Big/$$
$$\left(\sqrt{\left(-1-\upsilon l^2\right)(2\,t\,\upsilon l+i\,\upsilon g\,\upsilon l-2\,x\,\mathrm{Cos}[\theta l]-i\,\mathrm{Cos}[\theta g+\theta l]+2\,y\,\mathrm{Sin}[\theta l])^2}\right)+$$
$$\Bigg(\Bigg(\mathrm{Erf}\left[\frac{1}{2}\sqrt{\frac{(2\,i\,t\,\upsilon l+\upsilon g\,\upsilon l-2\,i\,x\,\mathrm{Cos}[\theta l]-\mathrm{Cos}[\theta g+\theta l]+2\,i\,y\,\mathrm{Sin}[\theta l])^2}{1+\upsilon l^2}}\right]$$
$$\sqrt{(2\,i\,t\,\upsilon l+\upsilon g\,\upsilon l-2\,i\,x\,\mathrm{Cos}[\theta l]-\mathrm{Cos}[\theta g+\theta l]+2\,i\,y\,\mathrm{Sin}[\theta l])^2}+$$
$$\left(2\,i+\mathrm{Erfi}\left[\frac{2\,t\,\upsilon l-i\,\upsilon g\,\upsilon l-2\,x\,\mathrm{Cos}[\theta l]+i\,\mathrm{Cos}[\theta g+\theta l]+2\,y\,\mathrm{Sin}[\theta l]}{2\sqrt{1+\upsilon l^2}}\right]\right)$$
$$(2\,t\,\upsilon l-i\,\upsilon g\,\upsilon l-2\,x\,\mathrm{Cos}[\theta l]+i\,\mathrm{Cos}[\theta g+\theta l]+2\,y\,\mathrm{Sin}[\theta l])\Bigg)$$
$$\Big(\mathrm{Cosh}\Big[1\Big/\left(8\left(1+\upsilon l^2\right)\right)\left(1+4\,x^2+4\,y^2+2\,\upsilon g^2+2\,\upsilon l^2+8\,t^2\,\upsilon l^2-8\,i\,t\,\upsilon g\,\upsilon l^2-\right.$$
$$4\,i\,x\,\mathrm{Cos}[\theta g]-8\,x\,(2\,t-i\,\upsilon g)\,\upsilon l\,\mathrm{Cos}[\theta l]+4\,x^2\,\mathrm{Cos}[2\,\theta l]-$$
$$4\,y^2\,\mathrm{Cos}[2\,\theta l]+8\,i\,t\,\upsilon l\,\mathrm{Cos}[\theta g+\theta l]+4\,\upsilon g\,\upsilon l\,\mathrm{Cos}[\theta g+\theta l]-$$
$$\mathrm{Cos}[2\,(\theta g+\theta l)]-4\,i\,x\,\mathrm{Cos}[\theta g+2\,\theta l]-4\,i\,y\,\mathrm{Sin}[\theta g]+16\,t\,y\,\upsilon l\,\mathrm{Sin}[\theta l]-$$
$$\left.8\,i\,y\,\upsilon g\,\upsilon l\,\mathrm{Sin}[\theta l]-8\,x\,y\,\mathrm{Sin}[2\,\theta l]+4\,i\,y\,\mathrm{Sin}[\theta g+2\,\theta l]\right)\Big]-$$
$$\mathrm{Sinh}\Big[1\Big/\left(8\left(1+\upsilon l^2\right)\right)\left(1+4\,x^2+4\,y^2+2\,\upsilon g^2+2\,\upsilon l^2+8\,t^2\,\upsilon l^2-8\,i\,t\,\upsilon g\,\upsilon l^2-\right.$$
$$4\,i\,x\,\mathrm{Cos}[\theta g]-8\,x\,(2\,t-i\,\upsilon g)\,\upsilon l\,\mathrm{Cos}[\theta l]+4\,x^2\,\mathrm{Cos}[2\,\theta l]-$$
$$4\,y^2\,\mathrm{Cos}[2\,\theta l]+8\,i\,t\,\upsilon l\,\mathrm{Cos}[\theta g+\theta l]+4\,\upsilon g\,\upsilon l\,\mathrm{Cos}[\theta g+\theta l]-$$
$$\mathrm{Cos}[2\,(\theta g+\theta l)]-4\,i\,x\,\mathrm{Cos}[\theta g+2\,\theta l]-4\,i\,y\,\mathrm{Sin}[\theta g]+16\,t\,y\,\upsilon l\,\mathrm{Sin}[\theta l]-$$
$$\left.8\,i\,y\,\upsilon g\,\upsilon l\,\mathrm{Sin}[\theta l]-8\,x\,y\,\mathrm{Sin}[2\,\theta l]+4\,i\,y\,\mathrm{Sin}[\theta g+2\,\theta l]\right)\Big]\Bigg)\Big/$$

$$(2\,i\,t\,\upsilon l + \upsilon g\,\upsilon l - 2\,i\,x\,\text{Cos}[\theta l] - \text{Cos}[\theta g + \theta l] + 2\,i\,y\,\text{Sin}[\theta l])\Bigg)$$

```
(* The final expression is considered in three parts,
        1) the constant part (k),
        2) e1: the first component that is included within the parentheses,
        3) e2: the second element

   This cell defines k and reformulates e1.
 *)

Clear[δ, γ, α1, exp1Arg, exp1, e1f, e1]
```

$$k = \frac{1}{8\sqrt{2}\,\pi^2\sqrt{1+\frac{1}{\upsilon l^2}}\,\upsilon l\,\text{Abs}[\text{Sec}[\theta l]]}\,\text{Sec}[\theta l]$$

```
// FullSimplify;
(* elements that are included in a number of parentheses *)
```

$$\gamma = 1 + \upsilon l^2;$$

$$\alpha 1 = (2\,i\,t\,\upsilon l - \upsilon g\,\upsilon l - 2\,i\,x\,\text{Cos}[\theta l] + \text{Cos}[\theta g + \theta l] + 2\,i\,y\,\text{Sin}[\theta l]);$$

```
(* The argument of the exponential within e1 -- Exp[] is not seen directly,
it is the sum of Cosh[.]-Sinh[.] *)
```

$$\text{exp1Arg} = 1/\left(8\left(1+\upsilon l^2\right)\right)\left(1 + 4\,x^2 + 4\,y^2 + 2\,\upsilon g^2 + 2\,\upsilon l^2 + 8\,t^2\,\upsilon l^2 + 8\,i\,t\,\upsilon g\,\upsilon l^2 + 4\,i\,x\,\text{Cos}[\theta g] - 8\,x\,(2\,t + i\,\upsilon g)\,\upsilon l\,\text{Cos}[\theta l] + 4\,x^2\,\text{Cos}[2\,\theta l] - 4\,y^2\,\text{Cos}[2\,\theta l] - 8\,i\,t\,\upsilon l\,\text{Cos}[\theta g + \theta l] + 4\,\upsilon g\,\upsilon l\,\text{Cos}[\theta g + \theta l] - \text{Cos}[2\,(\theta g + \theta l)] + 4\,i\,x\,\text{Cos}[\theta g + 2\,\theta l] + 4\,i\,y\,\text{Sin}[\theta g] + 16\,t\,y\,\upsilon l\,\text{Sin}[\theta l] + 8\,i\,y\,\upsilon g\,\upsilon l\,\text{Sin}[\theta l] - 8\,x\,y\,\text{Sin}[2\,\theta l] - 4\,i\,y\,\text{Sin}[\theta g + 2\,\theta l]\Big);$$

```
(* argument negative because Cosh[.]-Sinh[.],
Expand[] because will be passed to restExpImag[.] *)
exp1 = Exp[-exp1Arg // Expand];

(* e1f: reformulate e1 as a function.
   ϵ is the REAL part of the exponential, which is common in both e1,e2.
   ϵ will be kept as a variable until the very
 end so that Mathematica can simplified it easily *)
```

$$\text{e1f}[\alpha\_, \epsilon\_] = \frac{1}{\text{Sqrt}[-\delta]\,(-\alpha\,i)}\left(\left(\frac{\text{Sqrt}[\delta]}{\upsilon l}\,\upsilon l\,\text{Erf}\left[\frac{1}{2\,\text{Sqrt}[\delta]}\,\alpha\right]\right)\alpha + \left(2 - i\,\text{Erfi}\left[\frac{1}{2\,\text{Sqrt}[\delta]}\,(-\alpha\,i)\right]\right)(\text{Sqrt}[-\delta]\,(-i\,\alpha))\right)\epsilon\,\text{restExpImag}[\text{exp1}]$$

$$\text{e1} = \left(\left(\sqrt{\delta}\,\text{Erf}\left[\frac{1}{2}\sqrt{\frac{(\alpha)^2}{1+\upsilon l^2}}\right]\alpha + \left(2 - i\,\text{Erfi}\left[\frac{-(\alpha\,i)}{2\sqrt{\delta}}\right]\right)\sqrt{(-\delta)\,(-(\alpha\,i))^2}\right)(\text{Cosh}[\text{exp2Arg}] - \text{Sinh}[\text{exp2Arg}])\right)\Big/\left(\sqrt{(-\delta)\,(-(\alpha\,i))^2}\right);$$

$$\frac{1}{\alpha\sqrt{-\delta}}\,i\,\epsilon\left(-i\,\alpha\sqrt{-\delta}\left(2-\text{Erf}\left[\frac{\alpha}{2\sqrt{\delta}}\right]\right)+\alpha\sqrt{\delta}\,\text{Erf}\left[\frac{\alpha}{2\sqrt{\delta}}\right]\right)$$
$$\left(\text{Cos}\left[\frac{t\,\upsilon g\,\upsilon l^2}{1+\upsilon l^2}\right]-i\,\text{Sin}\left[\frac{t\,\upsilon g\,\upsilon l^2}{1+\upsilon l^2}\right]\right)\left(\text{Cos}\left[\frac{x\,\text{Cos}[\theta g]}{2\,(1+\upsilon l^2)}\right]-i\,\text{Sin}\left[\frac{x\,\text{Cos}[\theta g]}{2\,(1+\upsilon l^2)}\right]\right)$$
$$\left(\text{Cos}\left[\frac{x\,\upsilon g\,\upsilon l\,\text{Cos}[\theta l]}{1+\upsilon l^2}\right]+i\,\text{Sin}\left[\frac{x\,\upsilon g\,\upsilon l\,\text{Cos}[\theta l]}{1+\upsilon l^2}\right]\right)$$
$$\left(\text{Cos}\left[\frac{t\,\upsilon l\,\text{Cos}[\theta g+\theta l]}{1+\upsilon l^2}\right]+i\,\text{Sin}\left[\frac{t\,\upsilon l\,\text{Cos}[\theta g+\theta l]}{1+\upsilon l^2}\right]\right)$$
$$\left(\text{Cos}\left[\frac{x\,\text{Cos}[\theta g+2\,\theta l]}{2\,(1+\upsilon l^2)}\right]-i\,\text{Sin}\left[\frac{x\,\text{Cos}[\theta g+2\,\theta l]}{2\,(1+\upsilon l^2)}\right]\right)$$
$$\left(\text{Cos}\left[\frac{y\,\text{Sin}[\theta g]}{2\,(1+\upsilon l^2)}\right]-i\,\text{Sin}\left[\frac{y\,\text{Sin}[\theta g]}{2\,(1+\upsilon l^2)}\right]\right)$$
$$\left(\text{Cos}\left[\frac{y\,\upsilon g\,\upsilon l\,\text{Sin}[\theta l]}{1+\upsilon l^2}\right]-i\,\text{Sin}\left[\frac{y\,\upsilon g\,\upsilon l\,\text{Sin}[\theta l]}{1+\upsilon l^2}\right]\right)$$
$$\left(\text{Cos}\left[\frac{y\,\text{Sin}[\theta g+2\,\theta l]}{2\,(1+\upsilon l^2)}\right]+i\,\text{Sin}\left[\frac{y\,\text{Sin}[\theta g+2\,\theta l]}{2\,(1+\upsilon l^2)}\right]\right)$$

```
(* This cell reformulates e2. Comments would be similar to previous cell. *)
Clear[γ, α2, exp2Arg, exp2, e2f, e2]

γ = 1 + υl^2;
α2 = 2 i t υl + υg υl - 2 i x Cos[θl] - Cos[θg + θl] + 2 i y Sin[θl];
exp2Arg =
  1 / (8 (1 + υl^2)) (1 + 4 x^2 + 4 y^2 + 2 υg^2 + 2 υl^2 + 8 t^2 υl^2 - 8 i t υg υl^2 - 4 i x Cos[θg] -
      8 x (2 t - i υg) υl Cos[θl] + 4 x^2 Cos[2 θl] - 4 y^2 Cos[2 θl] + 8 i t υl Cos[θg + θl] +
      4 υg υl Cos[θg + θl] - Cos[2 (θg + θl)] - 4 i x Cos[θg + 2 θl] - 4 i y Sin[θg] +
      16 t y υl Sin[θl] - 8 i y υg υl Sin[θl] - 8 x y Sin[2 θl] + 4 i y Sin[θg + 2 θl]);
exp2 = Exp[-exp2Arg // Expand];
(*exp2 =  restExpImag[exp2]  restExpReal[exp2];*)
e2f [α_, ϵ_] = 1/α
    (Erf[1/(2 Sqrt[δ]) α] α + (2 i + Erfi[1/(2 Sqrt[δ]) (-α i)]) (-α i)) ϵ restExpImag[exp2];
e2 = ((Erf[1/2 Sqrt[(α)^2/δ]] Sqrt[(α)^2] + (2 i + Erfi[(-α i)/(2 Sqrt[δ])]) (-α i))
      (Cosh[expArg2] - Sinh[expArg2])) / α;
```

```
(* reformulate the expression EXCEPT its constant, k1 *)
Clear[expr, exprs]
expr[β1_, β2_, ϵ_] = e1f[β1, ϵ] + e2f[β2, ϵ];

(* Two-step simplification *)
exprs[β1_, β2_, ϵ_] = expr[β1, β2, ϵ] // Simplify;
exprs[β1_, β2_, ϵ_] = exprs[β1, β2, ϵ] = FullSimplify[exprs[β1, β2, ϵ]];


$Aborted

(* This is the output of the simplification *)
exprs[β1, β2, ϵ] =
```

$$\frac{1}{\sqrt{-\delta}}\epsilon\left(4\sqrt{-\delta}\,\text{Cos}\left[\frac{1}{1+\upsilon l^2}(\upsilon g\,\upsilon l-\text{Cos}[\theta g+\theta l])\,(t\,\upsilon l-x\,\text{Cos}[\theta l]+y\,\text{Sin}[\theta l])\right]-\left(\sqrt{-\delta}-i\sqrt{\delta}\right)\text{Erf}\left[\frac{\beta 1}{2\sqrt{\delta}}\right]\left(\text{Cos}\left[\frac{1}{1+\upsilon l^2}(\upsilon g\,\upsilon l-\text{Cos}[\theta g+\theta l])\,(t\,\upsilon l-x\,\text{Cos}[\theta l]+y\,\text{Sin}[\theta l])\right]-i\,\text{Sin}\left[\frac{1}{1+\upsilon l^2}(\upsilon g\,\upsilon l-\text{Cos}[\theta g+\theta l])\,(t\,\upsilon l-x\,\text{Cos}[\theta l]+y\,\text{Sin}[\theta l])\right]\right)\right);$$

```
(* k is manually restructured from its definition made way above *)
kRes = Sign[Sec[θl]] / (8 π^2 Sqrt[2 (υl^2 + 1)])

(* The coefficients of the exponential are also taken from above
   REAL part of the exponential only!!! *)
exponential =
  Exp[Together[-1/(8 (1 + υl^2)) - x^2/(2 (1 + υl^2)) - y^2/(2 (1 + υl^2)) - υg^2/(4 (1 + υl^2)) - υl^2/(4 (1 + υl^2)) -
      t^2 υl^2/(1 + υl^2) + 2 t x υl Cos[θl]/(1 + υl^2) - x^2 Cos[2 θl]/(2 (1 + υl^2)) + y^2 Cos[2 θl]/(2 (1 + υl^2)) - υg υl Cos[θg + θl]/(2 (1 + υl^2)) +
      Cos[2 (θg + θl)]/(8 (1 + υl^2)) - 2 t y υl Sin[θl]/(1 + υl^2) + x y Sin[2 θl]/(1 + υl^2)]];

(* Final otput of the convolution, plug exponential back in.
   The convolution with the odd-phased filter is identical to even-
 phased EXCEPT Cos[] is replaced with Sin[]. *)
exprFinalCos =
 kRes ϵ (4 Cos[1/(1 + υl^2) (υg υl - Cos[θg + θl]) (t υl - x Cos[θl] + y Sin[θl])]) /.
  ϵ → exponential
exprFinalSin =
  kRes ϵ (4 Sin[1/(1 + υl^2) (υg υl - Cos[θg + θl]) (t υl - x Cos[θl] + y Sin[θl])]) /.
   ϵ → exponential;
```

$$\frac{\text{Sign}[\text{Sec}[\theta l]]}{8\sqrt{2}\,\pi^2\sqrt{1+\upsilon l^2}}$$

$$\frac{1}{2\sqrt{2}\,\pi^2\sqrt{1+\upsilon l^2}}$$

$$e^{\frac{-1-4x^2-4y^2-2\upsilon g^2-2\upsilon l^2-8t^2\upsilon l^2+16t x\upsilon l\text{Cos}[\theta l]-4x^2\text{Cos}[2\theta l]+4y^2\text{Cos}[2\theta l]-4\upsilon g\upsilon l\text{Cos}[\theta g+\theta l]+\text{Cos}[2(\theta g+\theta l)]-16t y\upsilon l\text{Sin}[\theta l]+8x y\text{Sin}[2\theta l]}{8(1+\upsilon l^2)}}$$

$$\text{Cos}\left[\frac{1}{1+\upsilon l^2}(\upsilon g\,\upsilon l-\text{Cos}[\theta g+\theta l])\,(t\,\upsilon l-x\,\text{Cos}[\theta l]+y\,\text{Sin}[\theta l])\right]\text{Sign}[\text{Sec}[\theta l]]$$

```
energy = exprFinalCos^2 + exprFinalSin^2 // Simplify
```

$$\frac{1}{8\pi^4\,(1+\upsilon l^2)}\,e^{-\frac{1+4x^2+4y^2+2\upsilon g^2+2\upsilon l^2+8t^2\upsilon l^2-16t x\upsilon l\text{Cos}[\theta l]+4(x^2-y^2)\text{Cos}[2\theta l]+4\upsilon g\upsilon l\text{Cos}[\theta g+\theta l]-\text{Cos}[2(\theta g+\theta l)]+16t y\upsilon l\text{Sin}[\theta l]-8x y\text{Sin}[2\theta l]}{4(1+\upsilon l^2)}}\,\text{Sign}[\text{Sec}[\theta l]]^2$$

```
energyDraw = energy /. {Sign[Sec[θl]] → 1}
```

$$\frac{1}{8\pi^4\,(1+\upsilon l^2)}\,e^{-\frac{1+4x^2+4y^2+2\upsilon g^2+2\upsilon l^2+8t^2\upsilon l^2-16t x\upsilon l\text{Cos}[\theta l]+4(x^2-y^2)\text{Cos}[2\theta l]+4\upsilon g\upsilon l\text{Cos}[\theta g+\theta l]-\text{Cos}[2(\theta g+\theta l)]+16t y\upsilon l\text{Sin}[\theta l]-8x y\text{Sin}[2\theta l]}{4(1+\upsilon l^2)}}$$

C.3 Extrema analysis to tune Gabor motion energy

This section provides the Mathematica® notebook file that was used to complete the extrema analysis for tuning Gabor motion energy to the orientation and speed of a moving line.

```
(* Notebook for finding how to tune the energy to a particular motion.
   I.e. what should υg and θg should be set to in order to maximise energy.
     @author Evangelos Sariyanidi -- sariyanidi[at]gmail[dot]com *)

(* The energy function, copied from Simplify_Convolution3D.nb *)
energy = 1 / (8 π^4 (1 + υl^2))
    e^-((1+4 x^2+4 y^2+2 υg^2+2 υl^2+8 t^2 υl^2-16 t x υl Cos[θl]+4 (x^2-y^2) Cos[2 θl]+4 υg υl Cos[θg+θl]-Cos[2 (θg+θl)]+16 t y υl Sin[θl]-8 x y Sin[2 θl]) / (4 (1+υl^2)));

(* Compute the first order partial derivatives;
  find critical points. *)
dυg = D[energy, υg] // Simplify;
dθg = D[energy, θg] // Simplify;
criticPts = Reduce[dυg == 0 && dθg == 0 && Im[υl] == 0, {θg, υg}]
```

$$\left(\upsilon l \in \text{Reals} \;\&\&\; C[1] \in \text{Integers} \;\&\&\; \left(\left(\upsilon l \neq 0 \;\&\&\; \left(\theta g == -\frac{\pi}{2} - \theta l + 2\pi C[1] \;||\; \theta g == \frac{\pi}{2} - \theta l + 2\pi C[1]\right) \;\&\&\; \upsilon g == 0\right) \;||\; (\theta g == \pi - \theta l + 2\pi C[1] \;\&\&\; \upsilon g == \upsilon l) \;||\; (\theta g == -\theta l + 2\pi C[1] \;\&\&\; \upsilon g == -\upsilon l)\right) \;||\; \left(C[1] \in \text{Integers} \;\&\&\; \upsilon l == 0 \;\&\&\; \left(\theta g == -\frac{\pi}{2} - \theta l + 2\pi C[1] \;||\; \theta g == \frac{\pi}{2} - \theta l + 2\pi C[1]\right) \;\&\&\; \upsilon g == 0\right)\right)$$

```
(* The secon-order partial derivatives for the determinant of the Hessian;
   The determinant will be used to classify critical points,
i.e. are they maxima,minima etc. *)
dυg2 = D[energy, {υg, 2}] // Simplify;
dθg2 = D[energy, {θg, 2}] // Simplify;
dυgθg = D[dυg, θg] // Simplify;
det = dυg2 dθg2 - dυgθg^2 // Simplify;
det = det // FullSimplify;


(* Test the first set sof solutions *)
det /. {υg → υl, θg → (π - θl)};
FullSimplify[%]
```

$$\frac{e^{-\frac{4 x^2+4 y^2+2 \upsilon l^2+\left(2+8 t^2\right) \upsilon l^2+4 \left(-\upsilon l^2+(x-y) (x+y) \cos[2 \theta l]+4 t y \upsilon l \sin[\theta l]-4 x \cos[\theta l] (t \upsilon l+y \sin[\theta l])\right)}{2 \left(1+\upsilon l^2\right)}} \left(-1-\upsilon l^2\right)^2}{64 \pi^8 \left(1+\upsilon l^2\right)^5}$$

$$\frac{e^{-\frac{4 (t \upsilon l-x \cos[\theta l]+y \sin[\theta l])^2}{1+\upsilon l^2}}}{64 \pi^8 \left(1+\upsilon l^2\right)^3}$$

```
(* The expression above is always positive,
   if dυg2 is neg. for the same solution, then we have a local maximum *)
dυg2 /. {υg → υl, θg → (π - θl)}
Simplify[%]
FullSimplify[%]
```

$$\frac{e^{-\frac{4x^2+4y^2+8t^2\upsilon l^2-16t\,x\,\upsilon l\,\text{Cos}[\theta l]+4\left(x^2-y^2\right)\text{Cos}[2\theta l]+16t\,y\,\upsilon l\,\text{Sin}[\theta l]-8x\,y\,\text{Sin}[2\theta l]}{4\left(1+\upsilon l^2\right)}}\left(-1-\upsilon l^2\right)}{8\pi^4\left(1+\upsilon l^2\right)^3}$$

$$-\frac{e^{-\frac{2(t\upsilon l-x\,\text{Cos}[\theta l]+y\,\text{Sin}[\theta l])^2}{1+\upsilon l^2}}}{8\pi^4\left(1+\upsilon l^2\right)^2}$$

$$-\frac{e^{-\frac{2(t\upsilon l-x\,\text{Cos}[\theta l]+y\,\text{Sin}[\theta l])^2}{1+\upsilon l^2}}}{8\pi^4\left(1+\upsilon l^2\right)^2}$$

```
(* This concludes the sec. part. deriv. test (υg,θg) =
 (υl,π-θl) is a local maximum. *)
(* Let's test the remaining solutions *)
det /. {υg → 0, θg -> -π/2 - θl};
Simplify[%];
FullSimplify[%]
det /. {υg → 0, θg -> π/2 - θl};
Simplify[%];
FullSimplify[%]
```

$$-e^{-\frac{1+2x^2+2y^2+\upsilon l^2+4t^2\upsilon l^2+2(x-y)(x+y)\text{Cos}[2\theta l]+8t\,y\,\upsilon l\,\text{Sin}[\theta l]-8x\,\text{Cos}[\theta l](t\upsilon l+y\,\text{Sin}[\theta l])}{1+\upsilon l^2}}\Big/\left(64\pi^8\left(1+\upsilon l^2\right)^3\right)$$

$$-e^{-\frac{1+2x^2+2y^2+\upsilon l^2+4t^2\upsilon l^2+2(x-y)(x+y)\text{Cos}[2\theta l]+8t\,y\,\upsilon l\,\text{Sin}[\theta l]-8x\,\text{Cos}[\theta l](t\upsilon l+y\,\text{Sin}[\theta l])}{1+\upsilon l^2}}\Big/\left(64\pi^8\left(1+\upsilon l^2\right)^3\right)$$

```
dυg2 /. {υg → 0, θg → -θl} // FullSimplify
```

$$-e^{-\frac{2\left(x^2+y^2\right)+\left(1+4t^2\right)\upsilon l^2+2(x-y)(x+y)\text{Cos}[2\theta l]+8t\,y\,\upsilon l\,\text{Sin}[\theta l]-8x\,\text{Cos}[\theta l](t\upsilon l+y\,\text{Sin}[\theta l])}{2\left(1+\upsilon l^2\right)}}\Big/\left(8\pi^4\left(1+\upsilon l^2\right)^3\right)$$

```
det /. {υg → 0, θg -> -π/2 - θl};
(* The above expressions are always negative,
therefore according to sec. part. deriv. test
   they are saddle points and not extrema.*)
(* Therefore the local maximum (υg,θg) =
 (υl,π-θl) is also a global maximum. *)
```

Bibliography

[1] A. Adams, M. Mahmoud, T. Baltrušaitis, and P. Robinson. Decoupling facial expressions and head motions in complex emotions. In *Proc. Int'l Conf. Affective Computing and Intelligent Interaction*, pages 274–280. IEEE, 2015.

[2] E. H. Adelson and J. R. Bergen. Spatio-temporal energy models for the perception of motion. *The J. of the Optical Society of America*, 2(2):284–299, 1985.

[3] R. Adolphs. Recognizing emotion from facial expressions: Psychological and neurological mechanisms. *Behavioral and cognitive neuroscience reviews*, 1(1):21–62, 2002.

[4] R. Adolphs. Perception and emotion: How we recognize facial expressions. *Current Directions in Psychological Science*, 15(5):222–226, 2006.

[5] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[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*, 16(5):403–410, 2005.

[7] N. Ambady and R. Rosenthal. Thin slices of expressive behavior as predictors of interpersonal consequences: A meta–analysis. *Psychological Bulletin*, 11(2):256–274, 1992.

[8] N. Anantrasirichai, A. Achim, N. G. Kingsbury, and D. R. Bull. Atmospheric turbulence mitigation using complex wavelet-based fusion. *IEEE Trans. on Image Processing*, 22(6):2398–2408, 2013.

[9] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, and S. P. Zafeiriou. Feature-based lucas–kanade and active appearance models. *IEEE Transactions on Image Processing*, 24(9):2617–2632, 2015.

[10] A. Ashraf, S. Lucey, and T. Chen. Fast image alignment in the Fourier domain. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2480–2487, 2010.

[11] T. Baenziger, M. Mortillaro, and K. R. Scherer. Introducing the Geneva multimodal expression corpus for experimental research on emotion perception. *Emotion*, 12(5):1161–1179, 2012.

[12] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *Int'l J. computer vision*, 56(3):221–255, 2004.

[13] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 2. *Int'l J. Computer Vision*, 56(3):221–255, 2004.

[14] T. Baltrusaitis, N. Banda, and P. Robinson. Dimensional affect recognition using continuous conditional random fields. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 1–8. IEEE, 2013.

[15] T. Baltrušaitis, M. Mahmoud, and P. Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, volume 6, pages 1–6, 2015.

[16] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Automatic recognition of facial actions in spontaneous expressions. *J. of Multimedia*, 1(6), 2006.

[17] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numerical Analysis*, 8(1):141–148, 1988.

[18] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proc. European Conf. Computer Vision*, pages 404–417. 2006.

[19] C. M. Bishop and N. M. Nasrabadi. *Pattern Recognition and Machine Learning*. Springer, 2006.

[20] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 296–302, 1991.

[21] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the locals: Multi-way local pooling for image recognition. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 2651–2658, 2011.

[22] Y.-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *Int'l Conf. Machine Learning*, pages 111–118, 2010.

[23] S. Brahnam, C.-F. Chuang, R. S. Sexton, and F. Y. Shih. Machine assessment of neonatal facial expressions of acute pain. *Decision Support Systems*, 43(4):1242–1254, 2007.

[24] S. Butler, J. Tanaka, M. Kaiser, and R. Le Grand. Mixed emotions: Holistic and analytic perception of facial expressions. *J. of Vision*, 9(8):496, 2009.

[25] C. Cadieu and B. Olshausen. Learning intermediate-level representations of form and motion from natural movies. *Neural computation*, 24(4):827–866, 2012.

[26] A. J. Calder, A. M. Burton, P. Miller, A. W. Young, and S. Akamatsu. A principal component analysis of facial expressions. *Vision Research*, 41(9):1179–1208, 2001.

[27] A. Calder, G. Rhodes, M. Johnson, and J. Haxby. *Oxford Handbook of Face Perception*. Oxford University Press, 2011.

[28] E. Candes and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[29] O. Çeliktutan, S. Ulukaya, and B. Sankur. A comparative study of face landmarking techniques. *EURASIP J. Image and Video Processing*, 2013(1):13, 2013.

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

[31] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. Intensity rank estimation of facial expressions based on a single image. In *IEEE Int'l Conference on Systems, Man, and Cybernetics*, pages 3157–3162, 2013.

[32] Y. Chang, C. Hu, and M. Turk. Manifold of facial expression. In *Proc. IEEE Int'l Workshop on Analysis and Modeling of Faces and Gestures*, pages 28–35, 2003.

[33] S. Chew, P. Lucey, S. Lucey, J. Saragih, J. Cohn, I. Matthews, and S. Sridharan. In the pursuit of effective affective computing: The relationship between features and registration. *IEEE Trans. Systems, Man and Cybernetics – Part B*, 42(4):1006–1016, 2012.

[34] W.-S. Chu, F. De La Torre, and J. Cohn. Selective transfer machine for personalized facial action unit detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3515–3522, 2013.

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

[36] J. Cohn and K. L. Schmidt. The timing of facial motion in posed and spontaneous smiles. *Int'l J. of Wavelets, Multiresolution and Information Processing*, 02(02):121–132, 2004.

[37] 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*, pages 203–221, 2007.

[38] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[39] S. Cotter. Sparse representation for accurate classification of corrupted and occluded facial expressions. In *IEEE Int'l Conf. Acoustics Speech and Signal Processing*, pages 838–841, 2010.

[40] G. W. Cottrell, M. N. Dailey, C. Padgett, and R. Adolphs. Is all face processing holistic? The view from UCSD. *Computational, geometric, and process perspectives on facial cognition*, pages 347–396, 2001.

[41] M. Cox, S. Sridharan, S. Lucey, and J. Cohn. Least squares congealing for unsupervised alignment of images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[42] F. C. Crow. Summed-area tables for texture mapping. *ACM SIGGRAPH computer graphics*, 18(3):207–212, 1984.

[43] A. Cruz, B. Bhanu, and S. Yang. A psychologically-inspired match-score fusion mode for video-based facial expression recognition. In *Proc. Int'l Conf. Affective Computing and Intelligent Interaction*, pages 341–350, 2011.

[44] M. Dahmane and J. Meunier. Continuous emotion recognition using Gabor energy filters. In *Proc. Int'l Conf. Affective Computing and Intelligent Interaction*, pages 351–358, 2011.

[45] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[46] F. De la Torre, T. Simon, Z. Ambadar, and J. F. Cohn. Fast-FACS: a computer-assisted system to increase speed and reliability of manual FACS coding. In *Affective Computing and Intelligent Interaction*, pages 57–66. 2011.

[47] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Collecting large, richly annotated facial-expression databases from movies. *IEEE MultiMedia*, 2012.

[48] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *Proc. IEEE Int'l Conf. Computer Vision Workshops*, pages 2106–2112, 2011.

[49] A. Dhall, O. Ramana Murthy, R. Goecke, J. Joshi, and T. Gedeon. Video and image based emotion recognition challenges in the wild: Emotiw 2015. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 423–426. ACM, 2015.

[50] X. Ding, W.-S. Chu, F. De La Torre, J. F. Cohn, and Q. Wang. Facial action unit event detection by cascade of tasks. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 2400–2407, 2013.

[51] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1078–1085, 2010.

[52] R. Dooley. Microsoft glasses read your emotions. `http://www.neurosciencemarketing.com/blog/articles/microsoft-glasses.htm`, April 2015. Last accessed on Jun 10, 2015.

[53] E. Douglas-Cowie, N. Campbell, R. Cowie, and P. Roach. Emotional speech: Towards a new generation of databases. *Speech Communication*, 40(1):33–60, 2003.

[54] N. Dowson and R. Bowden. Mutual information for lucas-Kanade tracking (MILK): An inverse compositional formulation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (1):180–185, 2007.

[55] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal. Recurrent neural networks for emotion recognition in video. In *Proc. ACM Int'l Conf. Multimodal Interaction*, pages 467–474. ACM, 2015.

[56] P. Ekman, J. Campos, R. Davidson, and F. D. Waals. *Emotions Inside Out*, volume 1000 of *Annals of the New York Academy of Sciences*. New York Academy of Sciences, 2003.

[57] P. Ekman, W. Friesen, and J. Hager. *The Facial Action Coding System*. Weidenfeld and Nicolson, London, 2 edition, 2002.

[58] P. Ekman, W. Friesen, and J. Hager. The facial action coding system. *A Human Face*, 2002.

[59] S. Elaiwat, M. Bennamoun, and F. Boussaid. A spatio-temporal rbm-based model for facial expression recognition. *Pattern Recognition*, 49(C):152–161, 2016.

[60] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008.

[61] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. 2003.

[62] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *Int'l J. Computer Vision*, 5(1):77–104, 1990.

[63] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

[64] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[65] M. Glodek, S. Tschechne, G. Layher, M. Schels, T. Brosch, S. Scherer, M. Kächele, M. Schmidt, H. Neumann, G. Palm, and F. Schwenker. Multiple classifier systems for the classification of audio-visual emotional states. In *Proc. Int'l Conf. Affective Computing and Intelligent Interaction*, pages 359–368. 2011.

[66] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Prentice Hall, 2008.

[67] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *Int'l Conf. on Neural Information Processing*, pages 117–124, 2013.

[68] D. B. Grimes and R. P. Rao. Bilinear sparse coding for invariant vision. *Neural computation*, 17(1):47–73, 2005.

[69] T. Gritti, C. Shan, V. Jeanne, and R. Braspenning. Local features based facial expression recognition with face registration errors. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 1–8, 2008.

[70] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12):1080–1093, 2005.

[71] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807 – 813, 2010.

[72] A. Gudi, H. E. Tasli, T. M. den Uyl, and A. Maroulis. Deep learning based facs action unit occurrence and intensity estimation. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, volume 6, pages 1–5. IEEE, 2015.

[73] H. Gunes and B. Schuller. Categorical and dimensional affect analysis in continuous input: Current trends and future directions. *Image and Vision Computing*, 31(2):120 – 136, 2013.

[74] Z. Hammal and J. F. Cohn. Automatic detection of pain intensity. In *Proc. of ACM Int'l Conf. Multimodal Interfaces*, pages 47–52, 2012.

[75] Z. Hammal, J. F. Cohn, C. Heike, and M. L. Speltz. What can head and facial movements convey about positive and negative affect? In *Proc. Int'l Conf. Affective Computing and Intelligent Interaction*, pages 281–287. IEEE, 2015.

[76] S. Han, Z. Meng, A.-S. Khan, and Y. Tong. Incremental boosting convolutional neural network for facial action unit recognition. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 109–117. Curran Associates, Inc., 2016.

[77] M. Hansard and R. Horaud. A differential model of the complex cell. *Neural computation*, 23(9):2324–2357, 2011.

[78] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[79] D. J. Heeger. Normalization of cell responses in cat striate cortex. *Visual neuroscience*, 9(02):181–197, 1992.

[80] H. Hodson. Google glass, now in tune with your emotions. `https://www.newscientist.com/article/dn26153-google-glass-now-in-tune-with-your-emotions/`, September 2014. Last accessed on Jun 10, 2015.

[81] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. of Machine Learning Research*, 5:1457–1469, 2004.

[82] G. Huang, M. Mattar, H. Lee, and E. G. Learned-Miller. Learning to align from scratch. In *Adances in Neural Information Processing Systems*, pages 764–772, 2012.

[83] K.-C. Huang, S.-Y. Huang, and Y.-H. Kuo. Emotion recognition based on a novel triangular facial feature extraction method. In *Int'l Joint Conf. on Neural Networks*, pages 1–6, 2010.

[84] 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.

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

[86] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The J. of Physiology*, 160(1):106, 1962.

[87] 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*, 20(7):1237–1252, 2003.

[88] S. Jaiswal and M. Valstar. Deep learning the dynamic appearance and shape of facial action units. In *Proc. IEEE Winter Conf. on Applications of Computer Vision*, pages 1–8, 2016.

[89] L. A. Jeni, J. Girard, J. Cohn, and F. De La Torre. Continuous AU intensity estimation using localized, sparse facial feature space. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition Workshops*, 2013.

[90] 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*, pages 245–251, 2013.

[91] 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*, 44(2):161–174, 2014.

[92] B. Jiang, M. Valstar, and M. Pantic. Action unit detection using sparse appearance descriptors in space-time video volumes. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 314–321, 2011.

[93] 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*, pages 2983–2991, 2015.

[94] S. Kaltwang, O. Rudovic, and M. Pantic. Continuous pain intensity estimation from facial expressions. In *Int'l Symposium on Advances in Visual Computing*, pages 368–377. 2012.

[95] J.-K. Kamarainen, V. Kyrki, and H. Kalviainen. Invariance properties of Gabor filter-based features-overview and applications. *IEEE Trans. on Image Processing*, 15(5):1088–1099, 2006.

[96] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 46–53, 2000.

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

[98] H. Kaya, F. Gürpinar, S. Afshar, and A. A. Salah. Contrasting and combining least squares based learners for emotion recognition in the wild. In *Proc. ACM Int'l Conf. Multimodal Interaction*, pages 459–466, 2015.

[99] B.-K. Kim, J. Roh, S.-Y. Dong, and S.-Y. Lee. Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. *J. on Multimodal User Interfaces*, 10:173–189, 2016.

[100] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *Proc. British Machine Vision Conf.*, pages 275–1, 2008.

[101] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture-based approach to recognition of facial actions and their temporal models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(11):1940 –1954, 2010.

[102] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[103] S. Kumar, H. Azartash, M. Biswas, and T. Nguyen. Real-time affine global motion estimation using phase correlation and its application for digital image stabilization. *IEEE Trans. on Image Processing,*, 20(12):3406–3418, 2011.

[104] M. Kyperountas, A. Tefas, and I. Pitas. Salient feature and reliable classifier selection for facial expression classification. *Pattern Recognition*, 43(3):972–986, 2010.

[105] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers*, 42(3):300 –311, 1993.

[106] J. Le Moigne, N. S. Netanyahu, and R. D. Eastman. *Image registration for remote sensing*. Cambridge University Press, 2011.

[107] 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*, pages 33–48. 2014.

[108] R. Leal-Campanario, J. A. Barradas-Bribiescas, J. M. Delgado-García, and A. Gruart. Relative contributions of eyelid and eye-retraction motor systems to reflex and classically conditioned blink responses in the rabbit. *J. of Applied Physiology*, 96(4):1541–1554, 2004.

[109] Y. LeCun. Learning invariant feature hierarchies. In *Proc. European Conf. Computer Vision Workshops and Demonstrations*, volume 7583, pages 496–505. 2012.

[110] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[111] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Proc. IEEE Int'l Symposium on Circuits and Systems*, pages 253–256, 2010.

[112] S. Lee. Symmetry-driven shape description for image retrieval. *Image and Vision Computing*, 31(4):357 – 363, 2013.

[113] T. S. Lee. Image representation using 2D Gabor wavelets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

[114] 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*, pages 1–6, 2013.

[115] Y. Li, J. Chen, Y. Zhao, and Q. Ji. Data-free prior model for facial action unit recognition. *IEEE Trans. Affective Computing*, 4(2):127–141, April 2013.

[116] C.-T. Liao, H.-J. Chuang, C.-H. Duan, and S.-H. Lai. Learning spatial weighting for facial expression analysis via constrained quadratic programming. *Pattern Recognition*, 2013. (in press).

[117] S. Liao, W. Fan, A. Chung, and D.-Y. Yeung. Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features. In *Proc. IEEE Int'l Conf. Image Processing*, pages 665–668, 2006.

[118] C. Lindner, P. A. Bromiley, M. C. Ionita, and T. F. Cootes. Robust and accurate shape model matching using random forest regression-voting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 37(9):1862–1874, 2015.

[119] 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*, pages 180–184, 2014.

[120] G. C. Littleword, M. S. Bartlett, and K. Lee. Automatic coding of facial expressions displayed during posed and genuine pain. *Image and Vision Computing*, 27(12):1797–1803, 2009.

[121] G. Littlewort, J. Whitehill, T.-F. Wu, N. Butko, P. Ruvolo, J. Movellan, and M. Bartlett. The motion in emotion - a CERT based approach to the FERA emotion challenge. In *Proc. IEEE Int'l Conf. Automatic Face Gesture Recognition*, pages 897–902, 2011.

[122] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011.

[123] M. Liu, S. Li, S. Shan, and X. Chen. Au-aware deep networks for facial expression recognition. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 1–6. IEEE, 2013.

[124] 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*, pages 143–157. 2014.

[125] 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*, pages 1749–1756, 2014.

[126] P. Liu, S. Han, Z. Meng, and Y. Tong. Facial expression recognition via a boosted deep belief network. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1805–1812, 2014.

[127] F. Long, T. Wu, J. R. Movellan, M. S. Bartlett, and G. Littlewort. Learning spatiotemporal features by using independent component analysis with application to facial expression recognition. *Neurocomputing*, 93:126 – 132, 2012.

[128] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. of Computer Vision*, 60(2):91–110, 2004.

[129] 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*, 41(3):664–674, 2011.

[130] 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*, pages 94 –101, 2010.

[131] 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*, pages 94 –101, 2010.

[132] P. Lucey, J. Cohn, K. M. Prkachin, P. E. Solomon, S. Chew, and I. Matthews. Painful monitoring: Automatic pain monitoring using the UNBC-McMaster shoulder pain expression archive database. *Image and Vision Computing*, 30(3):197–205, 2012.

[133] S. Lucey, A. B. Ashraf, and J. Cohn. Investigating spontaneous facial action recognition through AAM representations of the face. In *Face Recognition Book*. Pro Literatur Verlag, 2007.

[134] S. Lucey, R. Navarathna, A. B. Ashraf, and S. Sridharan. Fourier lucas-kanade algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(6):1383–1396, 2013.

[135] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with Gabor wavelets. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 200–205, 1998.

[136] M. Mahoor, S. Cadavid, D. Messinger, and J. Cohn. A framework for automated measurement of the intensity of non-posed facial action units. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, pages 74 –80, 2009.

[137] M. H. Mahoor, M. Zhou, K. L. Veon, S. M. Mavadati, and J. F. Cohn. Facial action unit recognition with sparse representation. In *Proc. IEEE Int'l Conf. Automatic Face Gesture Recognition*, pages 336–342. IEEE, 2011.

[138] S. Mallat. Understanding deep convolutional networks. *arXiv preprint arXiv:1601.04920*, 2016.

[139] A. Martinez. Deciphering the face. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, pages 7–12, 2011.

[140] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (6):810–815, 2004.

[141] S. Mavadati, M. Mahoor, K. Bartlett, P. Trinh, and J. Cohn. DISFA: A spontaneous facial action intensity database. *IEEE Trans. Affective Computing*, 2013.

[142] D. McDuff, R. Kaliouby, T. Senechal, M. Amr, J. Cohn, and R. Picard. Affectiva-MIT facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 881–888, 2013.

[143] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schroder. The SEMAINE database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Trans. Affective Computing*, 3(1):5–17, 2012.

[144] A. Metallinou and S. Narayanan. Annotation and processing of continuous emotional attributes: Challenges and opportunities. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 1–8, April 2013.

[145] E. Meyers and L. Wolf. Using biologically inspired features for face processing. *Int'l J. of Computer Vision*, 76(1):93–104, 2008.

[146] A. Mollahosseini, D. Chan, and M. H. Mahoor. Going deeper in facial expression recognition using deep neural networks. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE, 2016.

[147] S. Moore and R. Bowden. Local binary patterns for multi-view facial expression recognition. *Computer Vision and Image Understanding*, 115(4):541–558, 2011.

[148] E. Muñoz, P. Márquez-Neila, and L. Baumela. Rationalizing efficient compositional image alignment. *Int'l J. of Computer Vision*, 112(3):354–372, 2015.

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

[150] D. Neth. *Facial configuration and the perception of facial expression*. PhD thesis, 2007.

[151] D. C. Neth and A. M. Martinez. A computational shape-based model of anger and sadness justifies a configural representation of faces. *Vision Research*, 50(17):1693–1711, 2010.

[152] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler. Deep learning for emotion recognition on small datasets using transfer learning. In *Proc. ACM Int'l Conf. Multimodal Interaction*, pages 443–449, 2015.

[153] M. Nicolaou, H. Gunes, and M. Pantic. Output-associative RVM regression for dimensional and continuous emotion prediction. *Image and Vision Computing*, 30(3):186–196, 2012.

[154] M. Nicolaou, V. Pavlovic, and M. Pantic. Dynamic probabilistic CCA for analysis of affective behaviour. In *Proc. European Conf. Computer Vision*, pages 98–111. 2012.

[155] J. Nicolle, K. Bailly, and M. Chetouani. Facial action unit intensity prediction via hard multi-task metric learning for kernel regression. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, volume 6, pages 1–6, 2015.

[156] J. Nicolle, V. Rapp, K. Bailly, L. Prevost, and M. Chetouani. Robust continuous prediction of human emotions using multiscale dynamic cues. In *Proc. ACM Int'l Conf. Multimodal Interaction*, pages 501–508, 2012.

[157] S. Nikitidis, A. Tefas, N. Nikolaidis, and I. Pitas. Facial expression recognition using clustering discriminant non-negative matrix factorization. In *Proc. IEEE Int'l Conf. Image Processing*, pages 3001–3004, 2011.

[158] S. Nikitidis, A. Tefas, N. Nikolaidis, and I. Pitas. Subclass discriminant nonnegative matrix factorization for facial image analysis. *Pattern Recognition*, 45(12):4080 – 4091, 2012.

[159] J. S. Oakland. *Statistical process control*. Routledge, 2007.

[160] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. In *Proc. Int'l Conf. Image and Signal Processing*, pages 236–243. 2008.

[161] M. Okade and P. K. Biswas. Video stabilization using maximally stable extremal region features. *Multimedia Tools and Applications*, 68(3):947–968, 2014.

[162] 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*, pages 74460S–74460S, 2009.

[163] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325, 1997.

[164] S. Oron, A. Bar-Hille, and S. Avidan. Extended Lucas-Kanade tracking. In *Proc. European Conf. Computer Vision*, pages 142–156. 2014.

[165] W. Pan, K. Qin, and Y. Chen. An adaptable-multilayer fractional Fourier transform approach for image registration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(3):400–414, 2009.

[166] M. Pantic and I. Patras. Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences. *IEEE Trans. Systems, Man and Cybernetics – Part B*, 36(2):433–449, 2006.

[167] 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*, page 5, 2005.

[168] 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*, page 5 pp., 2005.

[169] S. Park, G. Mohammadi, R. Artstein, and L.-P. Morency. Crowdsourcing micro-level multimedia annotations: The challenges of evaluation and interface. In *Proc. ACM Multimedia Workshops*, pages 29–34, 2012.

[170] I. Patras and M. Pantic. Particle filtering with factorized likelihoods for tracking facial features. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pages 97–102, 2004.

[171] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(11):2233–2246, 2012.

[172] N. Petkov and E. Subramanian. Motion detection, noise reduction, texture suppression, and contour enhancement by spatiotemporal Gabor filters with surround inhibition. *Biological Cybernetics*, 97(5-6):423–439, 2007.

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

[174] N. Pinto, D. D. Cox, and J. J. DiCarlo. Why is real-world visual object recognition hard? *PLoS computational biology*, 4(1):e27, 2008.

[175] C. Pramerdorfer and M. Kampel. Facial expression recognition using convolutional neural networks: State of the art. *arXiv preprint arXiv:1612.02903*, 2016.

[176] R. Ptucha and A. Savakis. Facial expression recognition using facial features and manifold learning. *Advances in Visual Computing*, pages 301–309, 2010.

[177] N. Qian, R. A. Andersen, and E. H. Adelson. Transparent motion perception as detection of unbalanced motion signals. i. psychophysics. *J. of Neuroscience*, 14(12):7357–7366, 1994.

[178] A. Rahman, D. Houzet, D. Pellerin, S. Marat, and N. Guyader. Parallel implementation of a spatio-temporal visual saliency model. *J. of Real-Time Image Processing*, 6(1):3–14, 2011.

[179] 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*, pages 2857–2864, 2011.

[180] A. Rav-Acha and S. Peleg. Lucas-Kanade without iterative warping. In *Proc. IEEE Int'l Conf. Image Processing*, pages 1097–1100, 2006.

[181] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.

[182] 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*, pages 808–822. 2012.

[183] F. Ringeval, B. Schuller, M. Valstar, S. Jaiswal, E. Marchi, D. Lalanne, R. Cowie, and M. Pantic. AV+EC 2015: the first affect recognition challenge bridging across audio, video, and physiological data. In *Proc. of Int'l Workshop on Audio/Visual Emotion Challenge*, pages 3–8, 2015.

[184] B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil. Exploiting unrelated tasks in multi-task learning. In *Int'l Conf. Artificial Intelligence and Statistics*, pages 951–959, 2012.

[185] R. Rubinstein, A. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proc. IEEE*, 98(6):1045–1057, 2010.

[186] O. Rudovic, V. Pavlovic, and M. Pantic. Multi-output Laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2634–2641, 2012.

[187] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 397–403, 2013.

[188] J. Saragih, S. Lucey, and J. Cohn. Face alignment through subspace constrained mean-shifts. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1034–1041, 2009.

[189] E. Sariyandi, H. Gunes, and A. Cavallaro. Probabilistic temporal subpixel registration for facial expression analysis. In *Proc. Asian Conf. Computer Vision*, pages 320–335. 2014.

[190] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 37(6):1113–1133, 2015.

[191] E. Sariyanidi, H. Gunes, and A. Cavallaro. Robust registration of dynamic facial sequences. *IEEE Trans. on Image Processing*, 2016 (accepted).

[192] E. Sariyanidi, H. Gunes, and A. Cavallaro. Learning bases of activity for facial expression recognition. *IEEE Trans. on Image Processing*, 2016 (major revision submitted on 11/11/2016).

[193] E. Sariyanidi, V. Dagli, S. C. Tek, B. Tunc, and M. Gökmen. Local Zernike Moments: A new representation for face recognition. In *Proc. IEEE Int'l Conf. Image Processing*, pages 585–588, 2012.

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

[195] A. Savran, H. Cao, M. Shah, A. Nenkova, and R. Verma. Combining video, audio and lexical indicators of affect in spontaneous conversation via particle filtering. In *Proc. ACM Int'l Conf. Multimodal Interaction*, pages 485–492, 2012.

[196] A. Savran, B. Sankur, and M. Taha Bilge. Regression-based intensity estimation of facial action units. *Image and Vision Computing*, 30(10):774–784, 2012.

[197] K. Scherer and P. Ekman. *Handbook of methods in nonverbal behavior research*. Cambridge Univ. Press, 1982.

[198] M. Schmidt. minfunc: unconstrained differentiable multivariate optimization in MATLAB, 2005.

[199] D. Schreiber. Robust template tracking with drift correction. *Pattern Recognition Letters*, 28(12):1483–1491, 2007.

[200] B. Schuller, M. Valstar, R. Cowie, and M. Pantic. AVEC 2012 - the continuous audio / visual emotion challenge. In *Proc. ACM Int'l Conf. Multimodal Interaction*, pages 361–362, 2012.

[201] B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic. AVEC 2011 - the first international audio/visual emotion challenge. In *Proc. Int'l Conf. Affective Computing and Intelligent Interaction*, pages 415–424. 2011.

[202] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In *Proc. ACM Conf. Multimedia*, pages 357–360, 2007.

[203] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost. Facial action recognition combining heterogeneous features via multikernel learning. *IEEE Trans. Systems, Man and Cybernetics – Part B*, 42(4):993–1005, 2012.

[204] T. Senechal, V. Rapp, and L. Prevost. Facial feature tracking for emotional dynamic analysis. In *Advanced Concepts for Intelligent Vision Systems*, volume 6915, pages 495–506. 2011.

[205] C. Shan, S. Gong, and P. W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803 – 816, 2009.

[206] C. Shan, S. Gong, and P. McOwan. Appearance manifold of facial expression. In *Computer Vision in Human-Computer Interaction*, volume 3766, pages 221–230. 2005.

[207] C. Shan, S. Gong, and P. McOwan. A comprehensive empirical study on linear subspace methods for facial expression analysis. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, page 153, 2006.

[208] 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 and Demonstrations*, pages 250–259. 2012.

[209] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.

[210] T. Simon, M. H. Nguyen, F. De la Torre, and J. Cohn. Action unit detection with segment-based SVMs. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2737–2744, 2010.

[211] G. Slager, E. Otten, T. Van Eijden, and J. Van Willigen. Mathematical model of the human jaw system simulating static biting and movements after unloading. *J. of Neurophysiology*, 78(6):3222–3233, 1997.

[212] P. Snape, A. Roussos, Y. Panagakis, and S. Zafeiriou. Face flow. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2993–3001, 2015.

[213] I. Sneddon, M. McRorie, G. McKeown, and J. Hanratty. The Belfast induced natural emotion database. *IEEE Trans. Affective Computing*, 3(1):32–41, 2012.

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

[215] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 824–832, 2015.

[216] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[217] J. W. Tanaka and I. Gordon. Features, configuration, and holistic face processing. *Oxford Handbook of Face Perception*, page 177, 2011.

[218] Y. Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.

[219] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proc. European Conf. Computer Vision*, pages 140–153, 2010.

[220] M. R. Teague. Image analysis via the general theory of moments. *The J. of the Optical Society of America*, 70(8):920–930, 1980.

[221] The MPLab GENKI Database: `http://mplab.ucsd.edu`.

[222] C. Theriault, N. Thome, and M. Cord. Dynamic scene classification: Learning motion descriptors with slow features analysis. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2603–2610, 2013.

[223] Y.-L. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.

[224] Y. Tong, J. Chen, and Q. Ji. A unified probabilistic framework for spontaneous facial action modeling and understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(2):258–273, 2010.

[225] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(10):1683 –1699, 2007.

[226] H. J. Towner. *Analysis of Feature Points and Physiological Data for Facial Expression Inference*. PhD thesis, 2007.

[227] V. J. Traver and F. Pla. Motion analysis with the radon transform on log-polar images. *J. of Mathematical Imaging and Vision*, 30(2):147–165, 2008.

[228] B. Tunç, V. Dağlı, and M. Gökmen. Class dependent factor analysis and its application to face recognition. *Pattern Recognition*, 45(12):4092–4102, 2012.

[229] M. Turk and A. Pentland. Eigenfaces for recognition. *J. of Cognitive Neuroscience*, 3(1):71–86, 1991.

[230] G. Tzimiropoulos, V. Argyriou, S. Zafeiriou, and T. Stathaki. Robust FFT-based scale-invariant image registration with image gradients. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(10):1899–1906, 2010.

[231] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3659–3667, 2015.

[232] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, and M. Pantic. Generic active appearance models revisited. In *Proc. Asian Conf. on Computer Vision*, pages 650–663. 2013.

[233] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Robust and efficient parametric face alignment. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1847–1854, 2011.

[234] M. Valstar, H. Gunes, and M. Pantic. How to distinguish posed from spontaneous smiles using geometric features. In *Procedings of the ACM Int'l Conf. Multimodal Interfaces*, pages 38–45, 2007.

[235] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2729–2736, 2010.

[236] M. Valstar and M. Pantic. Induced disgust, happiness and surprise: an addition to the MMI facial expression database. In *Proc. Int'l Conf. Language Resources and Evaluation Workshop on Emotion*, pages 65–70, 2010.

[237] M. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *IEEE Trans. Systems, Man and Cybernetics – Part B*, 42(1):28–43, 2012.

[238] M. Valstar, M. Pantic, Z. Ambadar, and J. Cohn. Spontaneous vs. posed facial behavior: Automatic analysis of brow actions. In *Proc. ACM Int'l Conf. Multimodal Interfaces*, pages 162–170, 2006.

[239] M. F. Valstar, T. Almaev, J. M. Girard, G. McKeown, M. Mehu, L. Yin, M. Pantic, and J. F. Cohn. FERA 2015 - second facial expression recognition and analysis challenge. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition Workshops*, volume 06, pages 1–8, 2015.

[240] M. Valstar, J. Gratch, B. Schuller, F. Ringeval, D. Lalanne, M. Torres Torres, S. Scherer, G. Stratou, R. Cowie, and M. Pantic. AVEC 2016: Depression, mood, and emotion recognition workshop and challenge. In *Proc. of the Int'l Workshop on Audio/Visual Emotion Challenge*, pages 3–10, 2016.

[241] M. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. Scherer. The first facial expression recognition and analysis challenge. In *Proc. IEEE Int'l Conf. Automatic Face Gesture Recognition*, pages 921–926, 2011.

[242] M. Valstar and M. Pantic. Combined support vector machines and hidden Markov models for modeling facial action temporal dynamics. In *Human-Computer Interaction*, volume 4796, pages 118–127. 2007.

[243] M. Valstar, B. Schuller, K. Smith, T. Almaev, F. Eyben, J. Krajewski, R. Cowie, and M. Pantic. AVEC 2014: 3D dimensional affect and depression recognition challenge. In *Proc. of the Int'l Workshop on Audio/Visual Emotion Challenge*, pages 3–10, 2014.

[244] M. Valstar, B. Schuller, K. Smith, F. Eyben, B. Jiang, S. Bilakhia, S. Schnieder, R. Cowie, and M. Pantic. AVEC 2013–the continuous audio/visual emotion and depression recognition challenge. In *Proc. ACM Int'l Conf. Multimodal Interfaces*, 2013.

[245] M. F. Valstar. *Timing is everything: A spatio-temporal approach to the analysis of facial actions*. Imperial College London, 2008.

[246] L. Van Der Maaten, E. Postma, and J. Van den Herik. Dimensionality reduction: a comparative. *J. of Machine Learning Research*, 10:66–71, 2009.

[247] 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*, 31(1):1–20, 2007.

[248] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743 – 1759, 2009.

[249] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using Gabor feature based boosted classifiers. In *IEEE Int'l Conf. Systems, Man and Cybernetics*, volume 2, pages 1692–1698, 2005.

[250] B. A. Wandell. *Foundations of vision.* Sinauer Associates, 1995.

[251] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.

[252] 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*, pages 3422–3429, 2013.

[253] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1385–1392, 2013.

[254] M. White. Parts and wholes in expression recognition. *Cognition & Emotion*, 14(1):39–60, 2000.

[255] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[256] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[257] 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.*, 31(4):65, 2012.

[258] 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*, pages 42–47, 2010.

[259] T. Wu, N. Butko, P. Ruvolo, J. Whitehill, M. Bartlett, and J. R. Movellan. Action unit recognition transfer across datasets. In *Proc. IEEE Int'l Conf. Automatic Face Gesture Recognition*, pages 889–896, 2011.

[260] X. Xiong and K. Qin. Linearly estimating all parameters of affine motion using radon transform. *IEEE Trans. on Image Processing*, 23(10):4311–4321, 2014.

[261] 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*, pages 532–539, 2013.

[262] L. Xu, J. Chen, and J. Jia. A segmentation based variational model for accurate optical flow estimation. In *Proc. European Conf. Computer Vision*, pages 671–684. 2008.

[263] J. Yan, Z. Lei, D. Yi, and S. Li. Learn to combine multiple hypotheses for accurate face alignment. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 392–396, 2013.

[264] 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*, pages 1849–1852, 2010.

[265] H. Yang and I. Patras. Sieving regression forest votes for facial feature detection in the wild. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1936–1943, 2013.

[266] P. Yang, Q. Liu, and D. N. Metaxas. Boosting encoded dynamic features for facial expression recognition. *Pattern Recognition Letters*, 30(2):132–139, 2009.

[267] P. Yang, Q. Liu, and D. N. Metaxas. Dynamic soft encoded patterns for facial event analysis. *Computer Vision and Image Understanding*, 115(3):456–465, 2011.

[268] P. Yang, Q. Liu, and D. Metaxas. Boosting coded dynamic features for facial action units and facial expression recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1 –6, 2007.

[269] S. Yang and B. Bhanu. Facial expression recognition using emotion avatar image. In *Proc. IEEE Int'l Conf. Automatic Face Gesture Recognition*, pages 866–871, 2011.

[270] A. Yao, J. Shao, N. Ma, and Y. Chen. Capturing au-aware facial features and their latent relations for emotion recognition in the wild. In *Proc. ACM Int'l Conf. Multimodal Interaction*, pages 451–458. ACM, 2015.

[271] Z. Yu and C. Zhang. Image based static facial expression recognition with multiple deep network learning. In *Proc. ACM Int'l Conf. Multimodal Interaction*, pages 435–442, 2015.

[272] A. Yüce, H. Gao, and J.-P. Thiran. Discriminant multi-label manifold embedding for facial action unit detection. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, volume 6, pages 1–6. IEEE, 2015.

[273] L. Zafeiriou, M. A. Nicolaou, S. Zafeiriou, S. Nikitidis, and M. Pantic. Probabilistic slow features for behavior analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 27(5):1034–1048, 2016.

[274] S. Zafeiriou and M. Petrou. Sparse representations for facial expressions recognition via l_1 optimization. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, pages 32–39, 2010.

[275] S. Zafeiriou, A. Papaioannou, I. Kotsia, M. Nicolaou, and G. Zhao. Facial affect "in-the-wild": A survey and a new database. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, 2016.

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

[277] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, 2004.

[278] L. Zhang, Y. Tong, and Q. Ji. Active image labeling and its application to facial action labeling. In *Proc. European Conf. Computer Vision*, pages 706–719. 2008.

[279] L. Zhang and D. Tjondronegoro. Facial expression recognition using facial movement features. *IEEE Trans. Affective Computing*, 2(4):219–229, 2011.

[280] L. Zhang and G. W. Cottrell. When holistic processing is not enough: Local features save the day. In *Proc. Annual Conf. of the Cognitive Science Society*, 2004.

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

[282] G. Zhao and M. Pietikäinen. Boosted multi-resolution spatiotemporal descriptors for facial expression recognition. *Pattern Recognition Letters*, 30(12):1117 – 1127, 2009.

[283] Q. Zhao, D. Zhang, and H. Lu. Supervised LLE in ICA space for facial expression recognition. In *Int'l Conf. Neural Networks and Brain*, volume 3, pages 1970–1975, 2005.

[284] 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*, 41(1):38–52, 2011.

[285] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. Metaxas. Learning active facial patches for expression analysis. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2562–2569, 2012.

[286] S. Zhu, C. Li, C. Change Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 4998–5006, 2015.

[287] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2879–2886, 2012.

[288] Y. Zhu, F. De la Torre, J. Cohn, and Y.-J. Zhang. Dynamic cascades with bidirectional bootstrapping for action unit detection in spontaneous facial behavior. *IEEE Trans. Affective Computing*, 2(2):79–91, 2011.