Towards Spatial and Temporal
Analysis of Facial Expressions in 3D
Data

Georgia Rajamanoharan

May 2015

Supervised by Dr. Stefanos Zafeiriou and Prof. Maja Pantic

Submitted in part fulfilment of the requirements for the degree of PhD in Computing
and the Diploma of Imperial College London. This thesis is entirely my own work, and,
except where otherwise indicated, describes my own research.
Copyright

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.
Abstract

Facial expressions are one of the most important means for communication of emotions and meaning. They are used to clarify and give emphasis, to express intentions, and form a crucial part of any human interaction. The ability to automatically recognise and analyse expressions could therefore prove to be vital in human behaviour understanding, which has applications in a number of areas such as psychology, medicine and security.

3D and 4D (3D+time) facial expression analysis is an expanding field, providing the ability to deal with problems inherent to 2D images, such as out-of-plane motion, head pose, and lighting and illumination issues. Analysis of data of this kind requires extending successful approaches applied to the 2D problem, as well as the development of new techniques. The introduction of recent new databases containing appropriate expression data, recorded in 3D or 4D, has allowed research into this exciting area for the first time.

This thesis develops a number of techniques, both in 2D and 3D, that build towards a complete system for analysis of 4D expressions. Suitable feature types, designed by employing binary pattern methods, are developed for analysis of 3D facial geometry data. The full dynamics of 4D expressions are modelled, through a system reliant on motion-based features, to demonstrate how the different components of the expression (neutral-onset-apex-offset) can be distinguished and harnessed. Further, the spatial structure of expressions is harnessed to improve expression component intensity estimation in 2D videos. Finally, it is discussed how this latter step could be extended to 3D facial expression analysis, and also combined with temporal analysis. Thus, it is demonstrated that both spatial and temporal information, when combined with appropriate 3D features, is critical in analysis of 4D expression data.
Acknowledgements

I would first of all like to thank my supervisors, Stefanos Zafeiriou and Maja Pantic, for all their help, advice and support over the last few years, and for providing the environment which has facilitated this work.

I would also like to thank my collaborators Daniel Rueckert and Lijun Yin for their helpful input to this work.

Many thanks go to members of the iBUG group for their help, particularly in providing the equipment and facilities that allowed me to complete this work, and for interesting and useful discussions. I would also like to thank a number of members of the BioMedIA group, with whom I was lucky to share an office, for their support.

Finally, I thank my husband and family, whose unending support has made this thesis possible.
# Contents

1 Introduction
   1.1 Motivation ......................................................... 1
   1.2 Facial Expression Analysis ........................................... 4
      1.2.1 Applications ................................................. 5
   1.3 Facial Action Coding System ......................................... 5
      1.3.1 Action Units .................................................. 7
      1.3.2 Intensities ................................................... 7
   1.4 2D vs 3D Data .................................................. 9
   1.5 Contributions ................................................... 11

2 Review of Facial Expression Analysis ........................................ 13
   2.1 Introduction .................................................... 13
   2.2 Preprocessing and Alignment ........................................ 15
      2.2.1 2D Facial Image Alignment .................................... 16
      2.2.2 3D Facial Geometry Alignment ................................ 17
   2.3 Feature Extraction .............................................. 20
      2.3.1 2D Static Features ........................................... 20
      2.3.2 2D Dynamic Features ....................................... 23
      2.3.3 3D Static Features .......................................... 24
      2.3.4 3D Dynamic Features ....................................... 32
   2.4 Feature Selection .............................................. 33
   2.5 Classification .................................................. 34
   2.6 Intensity Estimation ............................................. 36
   2.7 Temporal and Structural Modelling ................................... 37
   2.8 Databases ........................................................ 39
      2.8.1 2D Databases ................................................ 40
      2.8.2 3D Databases ................................................ 41
   2.9 Performance Measures ........................................... 47
## Contents

2.9.1 Classification Performance ........................................ 47  
2.9.2 Regression Performance .......................................... 49  
2.10 Summary and Challenges ............................................ 52  
  2.10.1 3D Features .................................................... 52  
  2.10.2 3D Dynamics ................................................... 53  
  2.10.3 AU Intensities .................................................. 53  
  2.10.4 3D Spontaneous Data ........................................... 54  

3 Binary Pattern Features for 3D Facial Action Unit Detection 55  
  3.1 Introduction ...................................................... 55  
  3.2 Facial Geometry Representations .................................. 57  
    3.2.1 Depth Map .................................................. 57  
    3.2.2 Facial Mesh Normals ...................................... 57  
    3.2.3 Azimuthal Projection Images ............................... 58  
  3.3 Preprocessing of Facial Representations .......................... 60  
    3.3.1 Gabor Filters ............................................... 60  
    3.3.2 Monogenic Filters ......................................... 61  
  3.4 Binary Pattern Features .......................................... 62  
    3.4.1 Local Depth Binary Patterns ............................... 63  
    3.4.2 Local Azimuthal Binary Patterns .......................... 63  
    3.4.3 Local Normal Binary Patterns .............................. 64  
    3.4.4 Local Phase Quantisers .................................... 66  
    3.4.5 Gabor Binary Patterns ..................................... 69  
    3.4.6 Monogenic Binary Patterns ................................. 70  
  3.5 Methodology ..................................................... 71  
    3.5.1 Feature Extraction ......................................... 72  
    3.5.2 Feature Selection ......................................... 75  
    3.5.3 Classification ............................................... 78  
    3.5.4 Feature Fusion .............................................. 79  
  3.6 Experiments ...................................................... 80  
    3.6.1 Parameter testing .......................................... 81  
    3.6.2 Full AU Performance ........................................ 85  
    3.6.3 Fusion Results .............................................. 90  
    3.6.4 Low Intensity AU Testing ................................... 92  
    3.6.5 Cross-Database Testing ..................................... 93  
    3.6.6 Summary and Discussion .................................... 94
## 4 Dynamic 3D Facial Expression Analysis

4.1 Introduction .................................................................................................................. 101
4.2 Feature Extraction ........................................................................................................ 103
   4.2.1 Motion Extraction ................................................................................................. 103
   4.2.2 Vector Projections ............................................................................................... 106
   4.2.3 Quad-Tree Decomposition .................................................................................. 109
   4.2.4 Features ................................................................................................................ 110
4.3 Modelling Expression Dynamics .................................................................................. 111
   4.3.1 Classification ........................................................................................................ 111
   4.3.2 Temporal Modelling ............................................................................................ 113
4.4 Experiments .................................................................................................................. 115
   4.4.1 Performance ......................................................................................................... 118
   4.4.2 Comparison to 2D ............................................................................................... 120
   4.4.3 Temporal Comparison ........................................................................................ 121
   4.4.4 Summary and Discussion .................................................................................... 124
4.5 Conclusions ................................................................................................................... 126

## 5 Facial Action Unit Intensity Estimation

5.1 Introduction ................................................................................................................... 127
5.2 Regression for Intensity Estimation ............................................................................. 129
   5.2.1 Feature Descriptor .............................................................................................. 129
   5.2.2 Feature Selection ................................................................................................. 130
   5.2.3 Regression ........................................................................................................... 131
5.3 An Intensity Markov Random Field Structure ........................................................... 133
   5.3.1 Face Region Model .............................................................................................. 135
   5.3.2 Tree Structures ................................................................................................. 137
   5.3.3 Model Parameters .............................................................................................. 138
   5.3.4 Intensity Inference ............................................................................................. 139
   5.3.5 Implementation ................................................................................................... 141
5.4 Experiments .................................................................................................................. 143
   5.4.1 Regression ........................................................................................................... 143
   5.4.2 Model Performance ............................................................................................. 145
   5.4.3 Comparison with Previous Work ........................................................................ 151
   5.4.4 Summary and Discussion .................................................................................... 151
5.5 Conclusions ................................................................................................................... 154
6 Conclusions

6.1 Contributions ........................................... 155
6.2 Findings .................................................. 156
6.3 Limitations and Future Work ......................... 157
   6.3.1 3D Data Storage and Processing ................. 157
   6.3.2 Intensity Estimation Dynamics .................... 158
   6.3.3 3D Spontaneous Expression Analysis ............ 158
   6.3.4 Deep Learning as an Alternative Approach .... 159
6.4 Summary .................................................. 160

Publications .................................................. 161

Appendix ....................................................... 163

Bibliography .................................................. 167
Acronyms

2DLBP  2D Local Binary Pattern.  80

2DLGBP  2D Local Gabor Binary Pattern.  85

3DLBP  3D Local Binary Pattern.  32

3DMM  3D Morphable Model.  19

AAM  Active Appearance Model.  16

AB  AdaBoost.  15

ADM  Annotated Deformable Model.  19

AEP  Azimuthal Equidistant Projection.  57

ANN  Artificial Neural Network.  15

APCI  Azimuthal Projection Component Image.  55

APDI  Azimuthal Projection Distance Image.  55

ASM  Active Shape Model.  17

AU  Action Unit.  2

BFSC  Basic Facial Shape Component.  28

BN  Bayesian Network.  15

CCNF  Continuous Conditional Neural Fields.  39

CCRF  Continuous Conditional Random Fields.  39
Acronyms

CLM Constrained Local Model. 17

CNF Conditional Neural Fields. 39

CORF Conditional Ordinal Random Field. 38

CRF Conditional Random Field. 38

DAG-SVM Direct Acyclic Graph SVM. 128

DBN Dynamic Bayesian Network. 15

DFT Discrete Fourier Transform. 67

EM Expectation-Maximisation. 35

ESC Expression Shape Component. 28

FACS Facial Action Coding System. 5

FFD Free Form Deformation. 18

GB GentleBoost. 34

GMC Gabor Mean Curvature. 88

GP Gaussian Process. 39

GSI Gabor Shape Index. 88

HGPP Histogram of Gabor Phase Pattern. 23

HMBP Histogram of Monogenic Binary Pattern. 23

HMM Hidden Markov Model. 15

HNC Haar Normal Components. 88

HOG Histogram of Oriented Gradient. 13

ICA Independent Component Analysis. 33
Acronyms

ICC  Intraclass Correlation Coefficient. 49
ICP  Iterative Close Point. 17
kNN  k-Nearest Neighbour. 34
LABP  Local Azimuthal Binary Pattern. 62
LAGBP  Local Azimuthal Gabor Binary Pattern. 62
LAMBP  Local Azimuthal Monogenic Binary Pattern. 63
LAPQ  Local Azimuthal Phase Quantiser. 62
LBP  Local Binary Pattern. 13
LBP-TOP  Local Binary Pattern-Three Orthogonal Planes. 24
LDA  Linear Discriminant Analysis. 33
LDBP  Local Depth Binary Pattern. 63
LDGBP  Local Depth Gabor Binary Pattern. 62
LDMBP  Local Depth Monogenic Binary Pattern. 63
LDP  Local Derivative Pattern. 23
LDPQ  Local Depth Phase Quantiser. 62
LGBP  Local Gabor Binary Pattern. 23
LGOBP  Local Gradient Orientation Binary Pattern. 23
LNBP  Local Normal Binary Pattern. 62
LNP  Local Normal Pattern. 85
LPQ  Local Phase Quantiser. 23
LPQ-TOP  Local Phase Quantiser-Three Orthogonal Planes. 24
LSCM  Least Squares Conformal Mapping. 19
**Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTP</td>
<td>Local Ternary Patterns.</td>
<td>23</td>
</tr>
<tr>
<td>ME-LBP</td>
<td>Multi Extended Local Binary Pattern.</td>
<td>32</td>
</tr>
<tr>
<td>MEM</td>
<td>Morphable Expression Model.</td>
<td>27</td>
</tr>
<tr>
<td>MHI</td>
<td>Motion History Image.</td>
<td>23</td>
</tr>
<tr>
<td>MP</td>
<td>Markov Process.</td>
<td>38</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov Random Field.</td>
<td>15</td>
</tr>
<tr>
<td>MS-LNP</td>
<td>Multi-scale Local Normal Patterns.</td>
<td>32</td>
</tr>
<tr>
<td>NMF</td>
<td>Non-negative Matrix Factorisation.</td>
<td>34</td>
</tr>
<tr>
<td>OF</td>
<td>Optical Flow.</td>
<td>13</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis.</td>
<td>15</td>
</tr>
<tr>
<td>PCC</td>
<td>Pearson Correlation Coefficient.</td>
<td>49</td>
</tr>
<tr>
<td>PNN</td>
<td>Probabilistic Neural Network.</td>
<td>35</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimisation.</td>
<td>18</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Squared Error.</td>
<td>49</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic.</td>
<td>47</td>
</tr>
<tr>
<td>ROC AuC</td>
<td>Receiver Operating Characteristic Area Under the Curve.</td>
<td>49</td>
</tr>
<tr>
<td>RVM</td>
<td>Relevance Vector Machine.</td>
<td>15</td>
</tr>
<tr>
<td>SFAM</td>
<td>Statistical Facial Feature Model.</td>
<td>26</td>
</tr>
<tr>
<td>SIFT</td>
<td>Scale Invariant Feature Transform.</td>
<td>21</td>
</tr>
<tr>
<td>SSVM</td>
<td>Structured Support Vector Machine.</td>
<td>153</td>
</tr>
<tr>
<td>STFT</td>
<td>Short-Term Fourier Transform.</td>
<td>66</td>
</tr>
</tbody>
</table>
Acronyms

**SVD** Singular Value Decomposition. 29

**SVM** Support Vector Machine. 15

**SVR** Support Vector Regressor. 15

**TAN** Tree-Augmented Naive Bayes. 38

**V-LGBP** Volume-Local Gabor Binary Pattern. 24

**VTK** The Visualisation Toolkit. 74
“His dress told her nothing, but his face told her things which she was glad to know.”

- A.A. Milne, Once on a Time

1.1 Motivation

As computers become more ubiquitous, entering more aspects of our everyday lives, the need for systems that can interact with us in natural ways is increasing. The technology that governs how we use computers has evolved rapidly in the last few years, and continues to change at a rapid pace, and the need for less intrusive methods for controlling interfaces is inevitable.

Automatic human behaviour understanding has attracted a great deal of interest over the past two decades, mainly because of its many applications spanning various fields such as psychology, computer technology, medicine and security. It can be regarded as the essence of next-generation computing systems as it plays a crucial role in affective computing technologies, deception detection systems, and patient-profiled personal well-being technologies [114].

Facial expressions are the most cogent, naturally preeminent means for humans to communicate emotions, to clarify and give emphasis, to signal comprehension disagreement, to express intentions and, in brief, to regulate interactions with the environment and other
1. Introduction

people [4]. These facts highlight the importance of automatic facial behaviour analysis, including facial expression analysis and facial Action Unit (AU) recognition, and justify the interest this research area has attracted, in the past twenty years [41, 55].

The objective of this thesis is to explore several aspects of facial expression analysis that have thus far received little attention. The majority of previous work in facial expression analysis has focussed on 2D intensity images or videos. However, 3D facial geometries offer the opportunity to capture a greater degree of information about the face, and overcome issues with lighting, pose, and texture changes. Therefore, features suitable for analysis of these geometries are developed for expression analysis, based on successful techniques applied to 2D images. These are applied to the problem of detecting the so-called components of expressions, facial AUs, which correspond to muscle movements in the face. The features are shown to be particularly helpful in detection of lower face AUs.

Harnessing expression dynamics have been shown to unlock important information about the emotional state [3] of the subject. They are essential for the recognition of complex states such as pain [192], and subtle emotions such as embarrassment [47]. 4D (3D+time) data allows the dynamics of expressions to be exploited along with the benefits of 3D facial geometries. A technique is developed for modelling the full dynamics of the expression (neutral-onset-apex-offset) from 3D expression data, for improved expression recognition.

Facial AUs intensity estimation is an area of expression analysis that has only recently become possible due to the publication of databases with suitable annotation [99, 94]. The ability to accurate estimate intensities would give a much richer level of information about the emotions of the subject [47], and help in analysis of complex states such as pain [125]. This is a challenging problem due to the complex ways in which the appearance of AUs can be greatly affected when they do not appear alone. However, AUs intensities display correlations, particularly in specific expression contexts, which can be harnessed for improved estimation. This area is explored.
1.2. Facial Expression Analysis

Figure 1.1: An overview of the field of facial expression analysis.
1. Introduction

1.2 Facial Expression Analysis

The problem of facial expression analysis can be tackled in a number of ways, as summarised in Fig. 1.1. Many researchers explore the task of whole expression analysis - either classification [28, 162] or intensity estimation [200, 73]. This is usually performed on the six universal expressions, which were found by Ekman et al. to be the only expressions recognised by all cultures across the world, and displayed in similar ways. These expressions are: Happy, Sad, Anger, Disgust, Fear and Surprise, and form the basis for a large number of facial expression databases that are currently available to researchers.

However, discrete expression recognition is limited, as the number of possible expressions is extremely wide, and training classification or regression systems for every possibility is unfeasible. As an alternative approach, some researchers have adopted a dimensional description of human emotion [106], where the range of expressions are characterised by a number of latent dimensions. Generally two dimensions are employed, as these are deemed to be sufficient to capture the majority of possible variation [34]. These are valence and arousal, which capture negativity versus positivity, and how active or inactive the emotion is, respectively. However, some works also employ a third dimension of dominance in order to capture more variation [56].

Facial AU recognition, either detection [142, 66], or intensity estimation [71, 144], is the final alternative approach. As AUs cover the full range of motions possible in the face, they provide a comprehensive system on which a higher level expression recognition system can be built. The main obstacle to robust AU recognition has been the time required to perform full AU annotation of databases, which has meant that until recently the majority of data did not contain this information. However, in recent years this has improved greatly, and so it is now possible to train facial AU recognition systems on a variety of data: posed, spontaneous, and both 2D intensity images and 3D facial geometries. However, the majority of work in this area has focussed on posed data, mostly 2D images, with little work so far conducted on either 3D AU detection or detection of AUs in spontaneous expression behaviour. Further, very little research has so far explored the area of AU intensity estimation.
1.3. Facial Action Coding System

1.2.1 Applications

Facial expression analysis can open up new approaches in a number of areas. The most obvious of these is in human-computer interaction, where expression recognition allows for computers that can understand, and react to, the users' emotions. Related to this is the inclusion into computer games, the ability for characters in the game to interact with the player in a more natural way, reacting not just to their input, but to how they are feeling during the course of play. Without expression recognition systems, this would be very difficult to infer.

Another area of interest in which expression analysis could play a vital role is in deception detection. Building accurate, non-intrusive lie detectors is an extremely challenging task, and it is known that a wide range of cues need to be employed, including facial expressions, to make this possible [46].

In addition, there are a number of medical applications in which facial expression analysis could be exploited beneficially. Pain detection from facial expressions has been shown recently to be possible, and a number of researchers have looked at this area [7, 71]. This could allow for better treatment, as doctors and medical practitioners can directly read the true level of pain, rather than relying on patients to tell them how they are feeling. Facial expressions, particularly during interactions, can also play an important role in diagnosis of medical conditions such as depression in adults, and autism in young children. Currently, these conditions require a long assessment process for diagnosis, but the employment of automatic systems for facial expression analysis could greatly reduce the time taken, and accuracy of, these processes [30].

1.3 Facial Action Coding System

The Facial Action Coding System (FACS) [57] provides a system for recognising the components of expressions in the face. It defines a set of facial AUs, which are directly linked to particular muscle movements, and cover all possible movements in the face. The aim of coding using this system is to determine, from the appearance of the face, which muscles are active in that expression. For example, Fig. 1.2 shows the muscle groups
1. Introduction

Figure 1.2: Examples of the underlying muscle movements for the upper face AUs (a) The muscle groups which activate the AUs (b) The resulting paths of movement for each of the AUs. Images from [57]

which underlie some of the upper face AUs. In Fig. 1.2a, the actual muscles are shown, while in Fig. 1.2b, the path along which these muscles pull in order to activate each AU are shown as black lines.

Because of the finite number of these AUs, FACS provides a comprehensive basis for expression recognition systems. Rather than a need to build systems able to recognise hundreds, or even thousands, of possible expressions, it is necessary only to detect which AUs are active, and from this higher-level emotional states can be inferred. An important aspect of FACS it that it is purely defined according to muscle movement, and is therefore not tied to any underlying emotional meaning or interpretation. This is an advantage of the system, as it means that it is completely objective. An AU and its appearance (alone or in combination) is comprehensively defined in the manual, and unaffected by
the expression context.

### 1.3.1 Action Units

The AUs defined in FACS are divided into several groups: upper face AUs (1-7 and 43-45), lower face AUs (9-18, 20, 22-24 and 28), lip parting and jaw opening AUs (25-27), eye positions (61-66) and miscellaneous AUs (8, 19 21 and 29-39) which cover jaw movements, nostril movement and some less common face motions. Most analysis systems focus on the first three of these groups, as they are the most common motions and generally of most interest for expression analysis. The upper face AUs concerns the forehead, eyes, and the top of the cheeks and nose, while the lower face concentrates on the rest of the nose and cheeks, and the mouth and chin. There are many more AUs possible in the lower face due to the number of muscles groups around the mouth. In Table 1.1 examples of some AUs are shown, along with their underlying muscle groups and appearance in the face. Here, all upper face AUs are included, while only a subset of the commonly recognised lower face AUs are shown.

The rules for coding according to the system are carefully defined in the FACS manual. In this manual, the appearance of every AU is described in great detail, both when expressed alone and combination with other AUs, in order to allow trained FACS coders to decide the full AU code for an expression image or video. Coding is a very time-consuming task, which is why producing fully coded facial expression data is a difficult process.

### 1.3.2 Intensities

As well as defining the appearance of each AU, the FACS manual also defines a range of intensities, and how to code these, for each AU. There are 5 ordinal levels of intensity defined, A-E, as shown in Fig. 1.3. Each of these levels actually defines a subrange of intensities. A is used to denote appearances that are barely visible, with only a trace of movement, and can only be coded by the most expert of FACS coders. Levels B-D are the most commonly displayed intensities, with B indicating a slight appearance of the AU, C either a marked or pronounced appearance, and D a severe or extreme appearance of the AU. Finally, level E is rarely used, but denotes the maximum possible demonstration of
1. Introduction

<table>
<thead>
<tr>
<th>AU</th>
<th>Description</th>
<th>Muscle Location</th>
<th>Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inner Brow Raiser</td>
<td>Inner forehead</td>
<td><img src="image1.png" alt="Image" /></td>
</tr>
<tr>
<td>2</td>
<td>Outer Brow Raiser</td>
<td>Outer forehead</td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>4</td>
<td>Brow Lowerer</td>
<td>Top of nose and eyebrows</td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>5</td>
<td>Upper Lid Raiser</td>
<td>Eyelids (tighten)</td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>6</td>
<td>Cheek Raiser</td>
<td>Outer circle around eye</td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
<tr>
<td>7</td>
<td>Lid Tightener</td>
<td>Inner circle around eye</td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>9</td>
<td>Nose Wrinkler</td>
<td>Top of nose</td>
<td><img src="image7.png" alt="Image" /></td>
</tr>
<tr>
<td>12</td>
<td>Lip Corner Puller</td>
<td>Up from lip corners in cheek</td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>14</td>
<td>Dimpler</td>
<td>Lip corners backwards</td>
<td><img src="image9.png" alt="Image" /></td>
</tr>
<tr>
<td>15</td>
<td>Lip Corner Depressor</td>
<td>Down from mouth corners</td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
<tr>
<td>20</td>
<td>Lip Stretcher</td>
<td>Lip corners out and down</td>
<td><img src="image11.png" alt="Image" /></td>
</tr>
<tr>
<td>22</td>
<td>Lip Funneler</td>
<td>Circle around mouth</td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>24</td>
<td>Lip Presser</td>
<td>Lips thin and shorten</td>
<td><img src="image13.png" alt="Image" /></td>
</tr>
<tr>
<td>28</td>
<td>Lips Suck</td>
<td>Lips not visible</td>
<td><img src="image14.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Table 1.1: The main upper and lower face AUs and their appearance.
the AU that the subject is capable of. It is still a range, but for only the absolute end of the spectrum.

For each intensity level of every AU, the appearance is clearly described in the FACS manual. However, inevitably, there will be some disagreement between coders, particularly when multiple AUs are active in a face. For this reason, it is common to have multiple coders code every image or video, and then use some form of voting or merging of their decisions to decide the final AU code.

### 1.4 2D vs 3D Data

The majority of work done in facial expression analysis thus far has employed 2D intensity images. Such systems exploit the texture of images to encode shape information about the face, and due to the large amount of 2D facial expression data available, have been able to tackle a number of aspects of facial expression analysis, achieving high performance on such datasets. However, most of these systems are highly sensitive to the recording conditions such as illumination, occlusions and other changes in facial appearance like makeup and facial hair. Furthermore, in most cases when 2D facial intensity images are used, it is necessary to maintain a consistent facial pose (preferably a frontal one) in order to achieve good recognition performance, as even small changes in the facial pose can reduce the system’s accuracy. In addition, single-view 2D analysis is unable to fully exploit the information displayed by the face as 2D video recordings cannot capture out-of-plane changes of the facial surface, or difficult to see changes. Hence, many 2D views must be utilised simultaneously if the information in the face is to be fully captured.

3D data, alternatively, provides a way to get around the problems inherent to 2D data. As this technique provides the full facial geometry, it is unaffected by lighting and other
1. Introduction

Figure 1.4: Examples of AU18 (Lip Pucker) and AU31 (Jaw Clencher) captured in both 2D and 3D. (a)-(d) AU18 2D nearly frontal view (e)-(h) AU18 3D reconstructed data (i)-(l) AU31 2D nearly frontal view (m)-(p) AU31 3D reconstructed data. The area of motion of the latter is shown in the circle.
1.5 Contributions

Textures on the face. In addition, pose and movement of the head can be easily accounted for. 3D data also ensures that all information in the face is captured including out-of-plane motion. For example, in the case of AU recognition, the subtle changes occurring in the depth of the facial surface are captured in detail when 3D data are used, something that does not happen with 2D data. For example, AU18 (Lip Pucker) is not easily distinguished from AU10+AU17+AU24 (Upper Lip and Chin Raising and Lip Presser) in a 2D frontal view video. In a 3D capture the action is easily identified. Examples of this as can be seen in Figs. 1.4a-1.4h. Similarly, AU31 (Jaw Clencher), can be difficult to detect in a 2D view, but is easily captured by the full 3D data. This comparison can be seen in Figs. 1.4i-1.4p. Recent advances in structured light scanning, stereo photogrammetry and photometric stereo have made the acquisition of 3D facial structure and motion a feasible task.

Facial expression recognition from 3D data is an expanding field, but so far the majority of work has focussed on static, posed data. Little work has so far explored the recognition of dynamic 3D expressions, and no work so far has performed recognition of 3D expressions in spontaneous facial expression behaviour.

1.5 Contributions

In each chapter of this thesis, several contributions are made to different aspects of automatic facial expression recognition and analysis from 3D and 4D data.

Chapter 3 The application of a number of binary pattern based features to 3D facial geometries for the purpose of AU detection is explored. The proposed features are based on methods successfully exploited on 2D intensity images. The different features are compared in a number of experiments on static data, and the benefits are demonstrated over the equivalent 2D features.

The introduction of a novel face image region selection method, based on individual feature selection, for application in AU detection. This method is shown to greatly reduce the computational expense of feature selection, whilst maintaining performance.
1. **Introduction**

The introduction of a classifier level feature fusion method that is demonstrated to successfully combine complementary features for overall improvement in performance, and outperform feature level fusion in computation expense and performance.

**Chapter 4** Exploitation of the expression dynamics of 4D (3D+time) data for improved expression recognition of the six basic expressions. 3D motion-based features are employed, and the temporal segments of the expressions (neutral-onset-apex-offset) are then modelled for recognition. This method is also shown to outperform the equivalent 2D method.

**Chapter 5** Investigation of the problem of AU intensity estimation, and application of a number of regression techniques to this problem. Application of a novel structured framework for improved intensity estimation through exploitation of correlations between AUs in spontaneous expression data. Work on intensity estimation is conducted on 2D data as the equivalent 3D is not yet available.

**Overall Structure** This thesis is structured as follows: Chapter 2 provides a full review of the field of facial expression analysis in both 2D and 3D data. This includes a summary of all aspects of a typical system, as well as the databases currently available and the measures of performance that are generally used in this field. In Chapter 3 the novel binary pattern based features are detailed and fully explored, along with region selection, for the purpose of AU detection. Chapter 4 describes how the dynamics of expressions are exploited for recognition purposes through temporal modelling. Then Chapter 5 details how the problem of AU intensity estimation is tackled, first by regression alone, and then by harnessing AU correlations through structural modelling. Finally Chapter 6 concludes the thesis and gives an overview of potential future work in this area.
Chapter 2

Review of Facial Expression Analysis

“The face is a picture of the mind with the eyes as its interpreter.”

- Cicero

2.1 Introduction

The human face contains a large number of muscles, and so is capable of a wide range of complex motions. As a result, facial expression analysis is a complex task which typically requires a number of processing stages. In this chapter an overview of a typical expression recognition system is provided, with the many techniques that have been proposed and utilised for each of the steps in this process reviewed in detail. ¹

The field of facial expression analysis has developed greatly in the last two decades, and there is now a large amount of literature in this area. The stages which make up a typically proposed system can be seen in Figure 2.1. Firstly, a preprocessing step is normally required to align and, in the case of 3D data, smooth, the images or meshes. This step can either be fully automatic, or require the input of manually annotated facial feature points. Feature extraction is then performed on each image, mesh or video frame, using a wide number of methods such as Gabor energy filters, Local Binary Patterns (LBPs), Histogram of Oriented Gradients (HOGs), and motion-based features extracted through Optical Flow (OF). The majority of methods produce a feature descriptor with a large

¹This background survey was published in large part as [SZPY12].
2. Review of Facial Expression Analysis

Figure 2.1: An overview of facial expression analysis systems.
number of dimensions and, as a result, require a feature selection or dimensionality re-
duction step to reduce the number of features. Techniques applied include Principal
Component Analysis (PCA), manifold learning algorithms, and AdaBoost (AB) feature
selection. Analysis can be then performed, the form of which depends on the partic-
ular application. For classification or detection of a discrete number of expressions or
AUs, methods include Support Vector Machines (SVMs) and Artificial Neural Networks
(ANNs). Alternatively, some works perform regression in order to perform expression or
AU intensity estimation, using methods such as Support Vector Regressors (SVRs), Rele-
vance Vector Machines (RVMs), or multiclass classifiers such as multiclass SVMs. Finally,
higher level modelling is applied in some works in order to harness the correlations either
between units (expressions or, more commonly, AUs), between frame temporal informa-
tion in videos, or both. Models applied include Bayesian Networks (BNs), either static
or Dynamic Bayesian Networks (DBNs), Hidden Markov Models (HMMs) and Markov
Random Fields (MRFs).

In the remaining sections in this chapter, the techniques proposed for each of these steps
are reviewed in more depth. The differences between techniques, particularly in feature
extraction, for 2D and 3D facial analysis are highlighted, as well as the common methods
applied for other steps in the facial analysis systems.

2.2 Preprocessing and Alignment

Accurate alignment and tracking methods are very important for facial expression sys-
tems, as the features extracted can often rely on areas of the face falling in the same
location between subjects, or finding the movement of particular points on the face. Face
detection, facial feature point extraction, and alignment, of 2D images is a largely solved
problem, with many robust systems proposed. In the case of 3D meshes, these tasks
are currently more difficult, so the majority of systems proposed so far rely on manually
annotated facial feature points to align the meshes. However, some methods have been
proposed for automatic alignment which are described here. Finally, dense correspon-
dence is sometimes required between face meshes in order to track the full motion of the
face between subjects or frames, and methods for this purpose have also been proposed.
2. Review of Facial Expression Analysis

2.2.1 2D Facial Image Alignment

Due to the nature of the features focussed on in this thesis, here only the problem of alignment of frontal facing, or same facing, images, in which the same set of facial points are visible in both images (i.e. there are no occlusions) is considered. Dealing with faces of different orientations in facial expression analysis is a more complex problem that is not included in this discussion.

The general aim in 2D facial image alignment is to identify a similarity or affine transform from the target image to the reference image. This can be calculated via a bounding box around the face, a number of facial feature points, or directly from the whole image. The simplest way to do this is to manually annotated a number of points, or box, in the image. But there have also been a number of automatic methods proposed for this problem that have been well used in facial expression analysis systems.

One of the most common methods employed for face detection, and thus facial alignment, is the technique that has become known as the Viola-Jones detector \cite{184}. This method uses an integral image representation to compute features, along with a learning algorithm based on AB to select features and create a set of classifiers. Classifiers are then combined in a cascade that discards the background and focuses on the promising face-like regions. The Viola-Jones detector has been used in many facial expression analysis systems \cite{78, 89, 26, 112}, typically to first detect the face location in an image before other more accurate facial point detectors are applied.

Active Appearance Models (AAMs) \cite{32} are a very widely applied method for extracting facial feature points, which can be used for alignment and facial feature point detection once a face has been detected. First proposed over 10 years ago, they remain the basis for state-of-the-art methods for face detection and point extraction. They employ a set of shape and texture basis vectors, derived from the variations in a training set of faces. The target image is then fitted to a linear combination of these bases through an optimisation algorithm. Since their initial introduction, many improvements have been proposed, such as coupled AAMs \cite{33} which use multiple views and models in combination, and the use of Bayesian AAMs \cite{2} in order to generalise the fitting to be person-invariant. AAMs
and their variants have been widely utilised in facial expression systems for alignment [7, 119, 93, 40]. However, the main constraint of the majority of AAM methods is that the models produced are person-specific, which means that they can not be applied to unseen individuals, and are mainly useful for tracking facial points throughout a video, after initialisation on the first frame. Active Shape Models (ASMs) [31] which were the predecessors to AAMs, were in the early years of facial analysis also applied to this problem. They employ only the shape bases to fit facial points to an image. However, recently their application in 2D methods has been replaced by AAMs, but they have been employed in 3D methods as will be described in Section 2.2.2.

Constrained Local Models (CLMs) [37] are an alternative model that is closely related to AAMs, which have also been employed in facial expression systems for face detection [26] and facial point detection [208, 64]. They employ additional shape constrained template searches during the fitting algorithm, which allows the models to be applied in person-independent applications.

Alternative methods apply direct alignment of similar frames, without the identification of facial feature points. One such method is the optimisation of image correlation coefficients based on the image gradients [176] which was employed for alignment before facial expression analysis in [67].

### 2.2.2 3D Facial Geometry Alignment

Iterative Close Point (ICP) [13] is an algorithm that has been widely used for 3D rigid alignment problems in 3D facial expression analysis [206, 53, 103]. The algorithm takes a source and target mesh, and then works by selecting either a subset or all of the points in each mesh. Then for each point in the target, it finds the closest point in the source, and aims to minimise the error between these points by applying a rigid transformation between the two meshes. This process is repeated until a threshold error is reached. Many variants of ICP have also been proposed [25, 158, 136].

However, this form of ICP only allows a rigid transformation, which does not find full correspondence of points between meshes of different individuals or when expression changes
2. Review of Facial Expression Analysis

occur. Additional methods are therefore required to perform a mapping or non-rigid transformation that produce full dense correspondence. In [5], a non-rigid version of ICP was proposed. This algorithm works by introducing a stiffness value which controlled the rigidity of the transformation that could be applied at each iteration. At the start this stiffness is given a high value, to force nearly rigid transformations, and then it is gradually reduced in order to allow progressively more non-rigid transformations to be applied as the iterations progress.

The depth map has been used itself, along with texture values at each coordinate and a Particle Swarm Optimisation (PSO) algorithm, in order to perform 3D alignment [69]. This work exploited a pyramidal approach based on the covariance matrix of a region, employing the PSO algorithm at each level to search for the corresponding point in the neighbourhood.

Free Form Deformations (FFDs) is another family of techniques used for non-rigid registration, first proposed in [135]. The idea is to deform an object by manipulating an underlying lattice of control points. The lattice is regular in the source mesh, and then deformed through an optimisation process in order to allow registration in the target mesh. B-spline interpolation of the deformation then models the motion of corresponding points between the two 3D meshes. This method was employed in [190] in order to fit a coarse face mesh model to the first frame in the sequence.

Harmonic maps [45] are a widely employed approach for 3D alignment. This method embeds a mesh from a manifold with disc topology into a planar graph through minimisation of the harmonic energy. This method is beneficial as it does not suffer from local minima, folding or clustering of the mesh, and is not affected by the resolution, smoothness or pose of the original 3D data. They were exploited in [190] in order to find dense correspondences after initial registration.

Conformal mappings is a technique that has been widely exploited in 3D alignment and tracking. A conformal mapping is a function that maps points in the mesh into a new domain, whilst preserving angles between edges in the mesh. This idea is used in order to produce 2D representations of the 3D data in [161, 130]. A circle pattern conformal
mappings are employed to convert the data into a 2D planar mesh. A generic model, also mapped to 2D using the same algorithm, is used for first coarse, and then fine, alignment and vertex correspondence. An alternative conformal mapping, Least Squares Conformal Mapping (LSCM), is applied in [188] for a similar purpose. Here AAMs are exploited to find features which allow a rough correspondence to be computed. LSCM is then applied to produce 2D planar meshes which were employed for dense correspondence. Harmonic maps, conformal mapping and LSCM are also used in [189].

ASMs, the predecessor to AAMs, though not employed in many recent 2D facial expression analysis systems, have been adopted in a number of 3D systems in order to perform tracking of facial features [175, 174]. The shape of the face is represented as a sequence of points which correspond to salient facial features. These points are formed from basis shapes computed from the principal components of a set of training faces, added to the mean of this training data.

3D Morphable Models (3DMMs), which will be described in Section 2.3.3, is an alternative method that is used in [105] to track 3D objects through image sequences. The model allows rigid motion in the form of translation and rotation away from the original mesh, plus non-rigid motion which is defined as a linear combination of basis vectors. The difference between one frame and the next was thus defined as being dependant on the change in motion parameters that allowed the target to be aligned with the current image. A matrix factorisation was found, allowing a large constant structure matrix to be precomputed off-line. A small time-varying motion matrix can be efficiently computed online and then used to update the motion parameters.

Finally, an alignment method which can be applied for finding dense correspondences is the Annotated Deformable Model (ADM) fitting proposed in [70]. The method fits the generic ADM in a novel 3D image and has been used for face recognition in the presence of facial expressions. As a preprocessing step an alignment method that uses spin images was applied in order to extract an initial correspondence between the data and the ADM. ICP is then employed, followed by refinement through the comparison of the z-buffer images for the model and data. Finally fitting is completed through iteratively deforming the generic model.
2. Review of Facial Expression Analysis

2.3 Feature Extraction

The feature extraction step is the main point at which facial expression analysis techniques in systems built to process 2D and 3D data differ greatly. In this thesis, 3D data and the techniques developed for 3D facial analysis are mainly focussed on. However, work on 3D data is necessarily highly influenced by the methods developed for processing 2D images, either through extensions of techniques, or even direct application of 2D methods to representations of the 3D data. And in addition, in Chapter 5, 2D methods are required for exploring AU intensity estimation due to the data available in this area. So for these reasons, first the techniques developed in 2D facial feature extraction are summarised. Then a more detailed description of the family of methods based on binary pattern operators is described, as this is the basis for Chapter 3. Finally, in the last two Sections, the methods that have been employed in 3D feature extraction for the purpose of facial expression analysis are explained in detail.

2.3.1 2D Static Features

2D static feature extraction approaches can mostly be divided into two categories of approaches: geometric features, which exploit the position or tracking of particular facial feature points, and appearance based features, that extract features from the whole image or regions of the face. One type of appearance based feature, LBP, is of particular interest, and so this is described in detail, as well as the range of variants that have been proposed in the literature.

Geometric features Geometric approaches, which concentrate on the shapes of particular facial components or the position of facial fiducial points, have been widely employed for expression analysis. The facial points are commonly automatically extracted through the use of AAMs and related methods, but can also be manually annotated in the dataset. These include the works in [76, 115, 116, 181], each of which track facial feature points, and use the movements of these points to find intermediate parameters. Some works extract the position of combinations of points through facial component models [169]. Alternatively, AAM coefficients can be used directly as features [146], and similarly, CLMs have also been used to extract shape features from the face [42, 91].
### Appearance-based features

Appearance methods are an alternative approach, that exploits features which make use of the whole face. They have been used throughout facial expression recognition work on 2D image sequences. These include the use of Gabor wavelets [172, 89] to produce feature representations of each frame that can then be used for classification. Gabor filters are a form of band-pass filters that have both scale and orientation parameters, which allow the extraction of different types of information from the image. The work in [216] is another example of work that used a set of Gabor wavelet coefficients, but this time facial features points were identified before applying the filters at only these locations. An alternative filter type, exploited in [209], are morphological operators which use the processes of dilation and erosion to highlight various facial characteristics useful for expression analysis.

An alternative feature descriptor that has been employed for facial expression recognition are LBPs, [149], as will be described in Section 2.3.1. HOG descriptors [38] are also a popular technique for feature extraction [85, 84]. These features are extracted by dividing the image into small regions, and forming a histogram of the gradient directions within each region. Finally, Scale Invariant Feature Transform (SIFT) features are another class of image features that selects key locations at maxima and minima of a difference of Gaussian function applied in the scale space, and then extracts descriptors at each of these points. They have been widely applied to extract features for facial expression analysis [59, 85].

### Local Binary Pattern Features

LBPs is a family of techniques that have been widely applied to the problem of feature extraction for facial expression recognition [87, 149, 201]. These features, proposed in [109], are a simple and fast method for encoding the image at each pixel. A neighbourhood is defined around each pixel with $P$ points. This neighbourhood can either be square or defined as regularly spaced points around a circle of radius $r$. The central pixel value is then used as a threshold in order to assign zero and one bits to the pixels in the neighbourhood, thus producing a binary number for that pixel. Hence, if $I$ is the image, then the LBP operator for a central point $I(x_c, y_c)$, with
2. Review of Facial Expression Analysis

Figure 2.2: The LBP operation with two different neighbourhoods. Top row employing a square neighbourhood, bottom row a circular neighbourhood, around the central pixel. (a)(d) Original pixel intensity values (b)(e) Difference in intensity values (c)(f) Resulting binary number.

$P$ neighbouring pixels $I(x_p, y_p)$ for $p = 0, ..., P - 1$ is defined as follows:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(I(x_p, y_p) - I(x_c, y_c))$$  \hspace{1cm} (2.1)

where

$$s(v) = \begin{cases} 
1 & \text{if } v \geq 0 \\
0 & \text{otherwise} 
\end{cases}$$  \hspace{1cm} (2.2)

An example of this operation on a circular neighbourhood is shown in Figs. 2.2d-2.2f. The binary numbers produced are then used to form histograms for regions in the image, and these are concatenated to form a feature descriptor for the image.

In the original paper [109], the authors proposed restricting the number of possible binary patterns, by taking only “uniform” patterns. These are defined as numbers for which there
are a maximum of two transitions from zero to one or vice versa, as the neighbourhood is travelled around.

**Binary Pattern Variants** Many variants of the original LBP operators have also been proposed. Local Gradient Orientation Binary Patterns (LGOBPs) [87, 86] which employ a second order binary number calculated by thresholding the gradients of the neighbourhood points, Local Derivative Patterns (LDPs) [212] which similarly use higher order binary results in a variety of directions, Local Ternary Patterns (LTPs) [163] which use three digit types, 0, 1, and −1 to further encode the intensities in the neighbouring points, and patch-based LBPs [193] which assign binary digits according to how similar neighbouring patches are to the central pixel patch.

Local Gabor Binary Patterns (LGBPs) [214, 208] and Histogram of Gabor Phase Patterns (HGPPs) [213] are other feature types based on the binary pattern methodology. The general approach here employs Gabor wavelets to first filter the images, before applying LBP algorithms to the magnitude or phase of the resulting coefficients. Similarly, Histogram of Monogenic Binary Patterns (HMBPs) [199] employ the monogenic signal, calculated using Gabor and Monogenic filters, to produce coefficient images which are then exploited to produce binary number histograms. Finally, Local Phase Quantisers (LPQs) [110] is a related feature type that uses the short-term Fourier transform to extract the phase information from the neighbourhood around each pixel in the image, and encode this with a binary number.

**2.3.2 2D Dynamic Features**

Alternatively, some works employ dynamic features which are described here. These are mainly motion based features, that track motion between all points in the face. Features that encode temporal information in other ways, and dynamic variants of LBPs, are then also described.

**Motion-based features** There are a number of methods for capturing motion between frames in 2D videos. A common method [126] is Motion History Image (MHI) [39], which set pixel intensity in an image according to the temporal history of that pixel. The higher
the intensity, the more recently that pixel changed. FFDs [135] use B-spline interpolation between a lattice of control points that are iteratively deformed in order to achieve the target image. They have been applied in a number of works to track motion between frames [75, 74, 126]. Other optical flow features have been employed in many works [202, 203, 137] to track movements between images or frames, often in combination with geometrical features [107]. An alternative method was employed in [88], where Gabor energy filters were used to encode the motion in image sequences.

**Encoding temporal information** Some works used features that implicitly encode the temporal information. These works do not aim to capture the motion, but rather to employ the information in neighbouring frames to describe the frame in question. In one example feature vectors representing images were embedded into a manifold in order for temporal analysis to be done [20]. The dynamics of the expression were then traced through the low-dimensional representation of the unfolded manifold. Two such methods were Local Binary Pattern-Three Orthogonal Planes (LBP-TOP) [217] and Local Phase Quantiser-Three Orthogonal Planes (LPQ-TOP) [67, 66], which will both be described in Section 2.3.1. Alternatively, multi-linear representations of the image sequence were used in [54] for classification.

**Dynamic Binary Patterns** LBPs, LPQs and LGBPs have also been extended to the dynamic problem. The LBP-TOP feature [217] is produced through taking three orthogonal planes from the spatial-time volume through each pixel and these are then employed to calculate binary patterns in each direction. LPQ-TOP [67] is an extension of this idea which again uses orthogonal planes in space and time, but this time with the LPQ algorithm, applied. Finally, Volume-Local Gabor Binary Pattern (V-LGBP) [197] extend the LGBP algorithm to the spatial-time volume.

### 2.3.3 3D Static Features

Several methods have been developed for the analysis of static 3D facial expressions. They use a range of different features for distinguishing between expressions or AUs, including characteristic distances, features from statistical models such as 3DMM and ASM
parameters, analysis of 2D representations and 3D variants of binary pattern features.

**Distance-based Features**  One of the most popular methods for feature extraction in 3D static faces is the use of characteristic distances between certain facial landmarks, calculated by taking into consideration the changes that occur in these due to facial deformations. This is comparable to the common geometric 2D methods that track fiducial points on the face. The BU-3DFE database provides the coordinates of 83 facial points in each mesh (as depicted in Fig. 2.9). Figs. 2.3a and 2.3b show some of the distance based features used in the literature.

The method developed in [153] uses six characteristic distances that are extracted from the distribution of 11 facial feature points from the given points in the BU-3DFE. In the method proposed in [154] a larger number of distances are extracted, corresponding to how open the eyes are, the height of the eyebrows, and several features that describe the position of the mouth. The work in [164] is another example of the use of facial points in the BU-3DFE. The distances between these points are normalised by Facial Animation Parameter Units. In addition, the authors use the slope of the lines joining these points, divided by their norms in order to produce unit vectors, as an additional set of features. Similarly, [83] uses six distances that are related to the movement of particular parts of the face, plus the angles of some slopes that relate to the shape of the eyes and mouth. In [165] a wider range of distances are calculated based on the given points in the BU-3DFE. The distances among all pairs of available 83 facial points were also used as features in [155, 156, 166].

Moreover, in [147] features were extracted by calculating the distances among all pairs of available face points. In addition, the surface curvature at each point in the mesh was classified as belonging to one of eight categories. The face was divided into triangles using a subset of the given facial points, and histograms were formed for each triangle of the surface curvature types. In [157] the authors used residues, which give both the magnitude and direction of the displacement of the given points in the BU-3DFE database, as features. A feature matrix was then formed by concatenating the different matrices in each of the three spatial directions in order to form one 2D matrix.
2. Review of Facial Expression Analysis

Figure 2.3: Different features based on the 83 given facial points in the BU-3DFE database. (a) Distance between particular given facial points used in [165, 164, 155, 156] (b) Distance and curvature features used in [147] (c) Circular patches around each facial point used in [95, 96]. Image (a) taken from [165], (b) taken from [147] (c) taken from [95].

Patch-based features Patches are another method that is widely employed for feature extraction in expression recognition systems. They are used to capture information about the shape of the face over a small local region around either every point in the mesh [186] or around landmarks or feature points [95, 96, 80]. Fig. 2.3c shows some of the patch based features used in the literature.

The authors in [186] computed a set of parameters for a smooth polynomial patch fitted to the local surface at each point in the mesh, which were subsequently used as inputs to rules that allowed the labelling of the surface at each point with primitives defining the type of curvature feature.

Alternatively, patches were found around landmarks in the 3D mesh in [95, 96]. These patches were used to define curves circling the points which show the level of the patch at those points. The square-root velocity function, that captures the shape of a curve, was calculated. The extracted values of the function were used to compute the necessary deformations between curves, and hence find a geodesic distance that represents the dissimilarity. The dissimilarity values were then added for all curves in a particular patch, in order to find one distance that represents the differences among patches.

Finally, [80] also found patches around landmarks in the face through fitting of the Sta-
2.3. Feature Extraction

Figure 2.4: Example of the morphable model fitted to a single image.

Statistical Facial Feature Model (SFAM), which is expressed as linear combinations of components of three different variations: shape, intensity and range value. These patches were then compared to the equivalent region from the six prototypical facial expressions through attempting to align them with ICP, and the distance between the patches after this process was employed as features.

Morphable models One of the most prominent methodologies for reconstructing the 3D facial surface from 2D facial images captured in unconstrained environments is the 3DMM methodology \[15, 16, 129, 18, 120, 48, 151, 17\] which also constitutes one of the most important recent developments in computational face modelling. Notably, the most well-known publicly available 3DMM is the one presented in \[15, 16\]. An example of this model fitted to an image can be seen in Fig. 2.4. This 3DMM is built from 3D laser scans of human faces that are put into dense correspondence using their pixel intensities and 3D shape information. A 3DMM uses a statistical representation of both the 3D shape and texture of the human face. The parameters of this, and related models, can be employed as features of 3D facial geometries.

For example, the Morphable Expression Model (MEM) was used in \[128\], and was able to model a range of different expressions for a particular individual. First the corresponding points in the expressive faces of a given individual were identified by reducing an energy function between points. Then the MEM was created by taking the principal components of the expressive faces of a person along with the average face, and performing a weighted
summation of these eigen-expressions to reconstruct a new face. Subsequently, the weights formed the representation of a new expression to be recognised.

Another alternative was the Basic Facial Shape Component (BFSC) model, that was able to model different identities with neutral expressions [53]. It was created as a linear combination of neutral faces. After mesh alignment using ICP, the BFSC was fitted to each mesh. Due to its nature BFSC is capable of modelling only neutral faces, therefore the subtraction of the depth map of BFSC from the depth map of the original aligned mesh provides the Expression Shape Component (ESC), that contains the expression information. This difference was then used to form the expression feature vector.

The SFAM was employed as an alternative type of morphable model in [219]. The model was fitted to the meshes under examination, and the parameters of the fitting were used to extract features. The intensity and range values were directly used, while the mean of the shape parameters was subtracted from this vector to extract a set of displacement features. In addition, the shape index was calculated from these parameters, and they were subsequently encoded via multi-scale LBPs to provide further descriptors. In [218], the SFAM was also used to extract features. Landmarks in the face were selected from the SFAM, either manually or automatically and features were extracted. The features consisted of the coordinates of the landmarks, as well as the morphology, texture and range parameters from local grids centered at the landmarks. LBP operators were applied to both texture and range parameters in order to encode the local properties around the landmarks. The changes in distances between some pairs of the landmarks were also calculated.

An elastically deformable bilinear 3D model was employed in [103]. This morphable model captures variations in both identity and expressions (as shown in Fig. 2.5). A prototypic facial surface model with neutral expression and average identity was fitted at the original point cloud data, and was later employed to establish correspondences between different faces. The model was fitted to the point cloud via landmarks that were identified on both the model and the point cloud. A subdivision surface between these points was created by minimising an energy function using an optimisation process. Once correspondence had been established, PCA was applied to find the principal components
2.3. Feature Extraction

Figure 2.5: Morphable model fitting in [103, 101, 102]. (a) The base mesh (b) The original surface data (c) The base mesh fitted to the surface. Images taken from [103].

of the base-mesh deformation, allowing any novel face to be written as a summation of these eigenmeshes. The face was then modelled via an asymmetric bilinear model based on these base meshes, and vector representations for each face model were formed during the fitting of the model. This feature extraction method was also used in [101, 102], though the optimal parameters were in this case found via differentiating the energy function, setting it equal to zero and then performing Singular Value Decomposition (SVD) to solve the acquired linear equations. The feature extraction method was also used in [104], though the energy function now required only the distances between the points to be minimised, rather than a bidirectional pull of both sets of points to one another.

2D Representations An alternative approach to the problem of feature extraction from 3D image sequences includes mapping the 3D data into a 2D representation. This representation can then be employed for either the division of the mesh area prior to the 3D features extraction, or for the direct application of traditional 2D techniques.

The depth map of the 3D facial meshes and the original $z$ values at each $x, y$ position were used as a 2D representation in [12]. The SIFT algorithm was then applied to extract features. The depth map was also employed in [185]. In this case the depth map is processed to achieve histogram equalisation over the image, and then Zernike moments
Differential geometry-based features were used to convert the 3D face data into a 2D representation in [142]. The acquired 2D representation was then analysed using a traditional 2D AU detection method. The 3D data was preprocessed to smooth and remove spikes, before mapping them into 2D curvature images, as seen in Figs. 2.6a and 2.6b, respectively. The curvature images were subsequently used to extract features via Gabor wavelets. Similarly, expressive maps were created in [108] through use of the ADM in order to extract a number of features such as the geometry, normals and local curvature. The maps described the discriminative nature of the points across the mesh, and could be used directly as features for classification purposes.

LSCM was used to form the 2D images from the 3D data in [141]. This was implemented through the method described in [140], which is an angle-preserving parameterisation method that produces consistent 2D images for 3D shapes. 2D elastic deformations were then used to estimate the correspondence between the image and a reference. Registration was performed using Gaussian image pyramids and multi-resolution meshes. Adaptive meshes were generated in order to smooth the contours and provide suitable point densities over the different parts of the face. These meshes were subsequently used for estimating the deformation via a non-rigid registration method that employs deformable triangular meshes that deform according to the stresses induced by the image matching errors.

The authors in [130] used conformal mappings to convert the 3D meshes to 2D planar meshes and find correspondences. An example of the conformal mapping representation found for the face data in Fig. 2.6c can be seen in Fig. 2.6d. Facial surface features on the mapped mesh were then labelled according to twelve primitives to form a facial expression label map. The labels were applied through estimation of the surface principal curvatures and directions by fitting a local facial surface and using the characteristics of the resulting Hessian matrix.

A combination of features derived from 2D texture, 2D and 3D surface, and 3D curvature were employed in [187]. These were determined from coefficients found from surface fitting using cubic functions and by computing the Gabor wavelet coefficients around landmarks.
Figure 2.6: 2D representations of 3D face data. (a) Original face range data (b) 2D curvature representation used in [142] (c) Original face range data (d) 2D conformal map representation used in [130]. Images (a)-(b) taken from [142].
2. Review of Facial Expression Analysis

in the face in order to compute moment invariants. 2D and 3D wavelet transforms were employed in [122] in order to extract multiscale features from the 3D face data.

**Binary Pattern Features** Some works have also begun to apply binary pattern descriptors to the 3D problem by proposing the application of the traditional LBP descriptor applied to the depth map of a facial mesh, as well as an extension of this, the 3D Local Binary Pattern (3DLBP) [61] and Multi Extended Local Binary Pattern (ME-LBP) [60], both of which use more information about the depth differences to encode the shape around each point. Additionally, Multi-scale Local Normal Patterns (MS-LNPs) have been applied for the problem of facial expression and action unit recognition in [82]. In this method, the angles of the normals were represented as intensities and the binary pattern algorithm applied to this representation.

2.3.4 3D Dynamic Features

A number of methods have also employed 3D dynamic features for facial expression analysis. In [21] feature points were tracked in order to capture the deformation of the 3D mesh during the expression. The tracking technique from [175] was used in both [173] and [174] to track the movement of landmarks in the face. The extracted information was subsequently used to determine the presence of different deformations in the face corresponding to particular AUs considering the change in measurements in different polygonal shapes represented by the landmarks. These methods were tested on a custom built database.

One of the first works that employed 3D motion-based features for facial expression analysis was presented in [206]. A deformable model was used to track the changes between frames, thus calculating motion vectors. The acquired motion vectors were then classified via an extracted 3D facial expression label map which was produced for each expression. This deformable model was also adapted to each frame in the image, and its changes were tracked in order to extract geometric features for use in [162].

A motion-based approach was also followed in [183]. The 3D facial meshes were mapped onto a uniform 3D matrix before subtracting the matrix corresponding to the neutral state for that subject. In that way a flow matrix was produced, showing the movement
appearing due to the expression evolvement through time. The Fourier Transform was subsequently applied to this matrix, and the rows of the resulting spectral matrix were concatenated to form a feature vector representing the expression.

Facial level curves were used in [79] to extract spatio-temporal features which were subsequently used to analyse 3D dynamic expression sequences. These level curves were acquired from each frame by extracting the points that lay at a particular height on the face for different levels, after applying alignment and cropping. Features were then extracted by comparing the curves across frames using Chamfer distances. The features extracted from the previous, current and next frames were also considered for each frame in order to exploit temporal information.

2.4 Feature Selection

Feature selection and dimensionality reduction techniques are exploited in a number of systems in order to reduce the size of the feature descriptors prior to classification or regression. This reduces the computational expense of algorithms, and also allows a stronger focus on the most discriminative features or variations in the data, which can improve performance. A number of different methods have been applied in both 2D and 3D systems.

PCA [68] is a widely used technique for dimensionality reduction that has been employed in large number of both 2D and 3D facial expression recognition methods [180, 168, 155, 156, 147, 19]. It uses the covariance of the feature descriptor data to find the largest variations, which then allow reconstruction of the data from a much smaller number of bases. In some cases, Linear Discriminant Analysis (LDA), which will be described in Section 2.5, can be subsequently applied to create a discriminant subspace [155, 156, 206]. LDA has also been employed on its own for dimensionality reduction [127]. Further, Independent Component Analysis (ICA), which employs higher order statistics than PCA, has been employed for dimensionality reduction [90]. Other manifold learning techniques, such as Isomap [167], have also been applied in facial expression analysis systems [24].

Alternative methods explicitly choose the most discriminative features from the descriptor.
2. Review of Facial Expression Analysis

**AB** is a feature selection method that uses weak classifiers in a boosting algorithm which iteratively increases the focus on difficult to classify examples. It has been used in many facial analysis methods [10, 88, 172, 90, 112] for feature selection. **GentleBoost (GB)**, a variation of **AB** that uses a different update rule to improve stability [52], has also been employed for feature selection in a number of works [74, 75, 177, 208].

Discriminant measures have also been exploited to determine the features that should be considered for classification. A number of measures have been employed to determine the discriminative power of the feature vectors: the Chi square statistic [148], the Fisher criterion [155, 156], and the Kullback-Leibler divergence measure [165]. Finally, the normalised cut-based filter algorithm, that aims to represent this discriminative ability with as few features as possible, was used in [147] prior to the application of **PCA**. Similarly, **Non-negative Matrix Factorisation (NMF)** is an alternative approach that splits the feature matrices into non-negative components, each of which can be employed separately. This method has been employed in [64] in order to break down the features into components that can then be selected from based on discriminative power.

Finally, genetic algorithms have also been employed for the purpose of feature selection [195, 43]. These algorithms mimic natural selection, using inheritance and mutation to generate examples iteratively, selecting the fittest at each stage to continue the process.

### 2.5 Classification

Several classification techniques have been used facial expression analysis systems, either for classification of expressions or **AUs**. Some of the earliest works simply applied predefined rules about the movement of facial points, or the behaviour of features, in order to make a decision on expressions [173, 174] or **AUs** [117]. Other research applied rules to mid-level parameters determined from the actual features [115, 116].

Alternatively, linear classifiers [108] have been applied to carefully extracted features for expression classification. **LDA**, where features are projected into a subspace which maximises the clustering of classes in the training set, has also been widely applied [186, 130, 147]. **k-Nearest Neighbour (kNN)** classification, where examples are simply classified
according to their nearest neighbours, is another technique employed [187, 183], as well as more complex clustering algorithms [128] and Expectation-Maximisation (EM) [191]. Finally, manifold learning techniques, such as Isomap [187, 24], have also been applied to facial expression analysis problems.

As well as being employed for feature selection, the AB algorithm has been utilised for classification itself [172, 165, 142, 95, 96]. GB has also been employed for classification in [74, 75, 89].

SVMs have been very widely used in facial expression recognition systems, and are possibly the most popular method of classification employed in this field. SVMs is a supervised classification method that finds a decision boundary that maximises the perpendicular distance between itself and a set of chosen support vectors which are the closest examples from each class. Kernels can be used to implicitly project the examples into higher dimensional spaces in order to provide non-linear classification. They have been exploited for a number of tasks: to distinguish between whole expression labels [42, 65, 91, 164, 185, 186, 142, 157, 53, 95, 12, 96, 147, 80], detect full expression dynamics [88], detect particular AUs [208, 146], analyse facial action unit temporal segments [181, 10, 66], or a mixture of these purposes [76, 77].

Bayes classifiers are simple probabilistic classifiers built on Bayes’ theory, that use a model to make a decision about the class based on individual features. They have been applied for classification of expressions [186, 103, 102, 104, 141, 123], and AU detection [142] in a small number of works.

ANN classifiers constitute another popular approach for classification of expressions [153, 168] and AUs [169]. They use a number of layers of connected neurons that produce outputs from a number of inputs. These have been extended to Probabilistic Neural Networks (PNNs), where the final layer produces probability vectors for the possible classes rather than a discrete decision. They have been employed in [154, 156, 166, 83].
2.6 Intensity Estimation

Some recent works have attempted intensity estimation, either of whole expressions or of AUs. Expression intensity estimation has most commonly been performed on the six basic expressions as this is the most widely available data. Little work has so far been conducted on AU intensity estimation thus far. As the task of FACS coding data with intensities is very time consuming, it is only recently that databases containing this kind of annotation have begun to be released, and even more recently that 3D databases have been produced which contain AU intensity coding.

A variety of regression and ranking methods have been employed for the tasks of intensity estimation. One simple technique is to simply take the confidence values of the SVM [11, 89] or AB classifiers [58] as direct indication of intensity. These techniques have been employed with some success for AUs intensities.

One method that has been employed for expression intensity estimation is RankBoost [51], which is closely related to the feature selection and classification method, AB. This algorithm builds a cascade of weak learners, chosen to rank the training examples according to intensity [200].

An alternative method is to employ multiple binary SVM classifiers, trained on each intensity as a separate class, to form a multiclass classifier. This approach has been employed for AUs intensity estimation [97, 98]. This can be done a number of ways, including majority voting [165], where the final classification is taken as the answer given by the majority of the binary classifiers.

Alternatively, direct regression based techniques have also been employed. SVRs, which are the regression equivalent of SVMs and minimise the margin between a central prediction function and support vectors in the training examples, have been applied in a number of works [65, 144, 64] for AU intensity estimation. Finally, RVMs [170] are a probabilistic alternative to SVMs that avoid the need for parameter optimisation, and were employed in [71].
2.7 Temporal and Structural Modelling

Temporal modelling consists of harnessing the changes over time in a video or number of frames, possibly including dividing of the expression or AU into onset-apex-offset segments. Alternatively, structural modelling consists of harnessing links between different components in an individual image, which in the field of facial expression analysis are typically between AUs. Many works in the field of facial expression analysis have aimed to harness these types of information in order to improve on recognition or intensity estimation rates. The majority of methods have employed graphical models of various kinds for these purposes, several of which can be see in Fig. 2.7.

BNs, as shown in Fig. 2.7a, have been employed in a number of works [145]. They aim to model the directional dependencies between elements through the use of conditional probability distributions. A particular form, the Tree-Augmented Naive Bayes (TAN), which uses only tree-based networks, has been employed [28, 29] to model the dependencies between the different features in the facial images. An extension of the basic BN are DBNs, as seen in Fig. 2.7b. These model each time-slice with a static BN, but then also allow dependencies to be added from one time-slice to the next, between elements. They have been employed in a number of works in order to harness correlations between AUs both within, and between, frames [216, 172, 84].

An alternative model regularly used for dynamic facial expressions analysis is the HMM, a tool best known for its use in speech processing. These model the observable output as dependent on the states of hidden variables, which in this case generally represent the different temporal segments of the expression or AU. The hidden variables transition according to a defined Markov Process (MP), an example of which can be seen in Fig. 2.7c, and the HMM itself can be seen in Fig. 2.7d. Several methods use HMMs for temporal modelling [28, 203, 79] of expression dynamics or to model the dynamics of AUs [181, 75]. A variation of simple HMMs is the use of 2D spatio-temporal HMMs, which are employed in [162] to model both the spatial and temporal relationships in the features.

MRFs and Conditional Random Fields (CRFs) are two related models that have been widely used in computer vision problems. MRFs uses structure to model undirected
2. Review of Facial Expression Analysis

Figure 2.7: Some of the graphical models that have been employed in facial analysis systems (a) Bayesian Network (b) Dynamic Bayesian Network (c) Markov Process (d) Hidden Markov Model (e) Markov Random Field (f) Conditional Random Field
dependencies between hidden variables on which the observable nodes are dependant, and CRFs then use a higher level variable which adds an additional dependency to all hidden nodes. Examples can be seen in Figs. 2.7e and 2.7f respectively. Though they themselves have not been employed for facial expression analysis thus far in the literature, an extension of CRFs, Conditional Ordinal Random Fields (CORFs) that exploit ordinal states, have been employed to model the temporal segments of both expressions [132] and AUs [131], as well as expression intensites [73] and AU intensities [133]. CRFs have also been extended to deal with continuous cases to produce the Continuous Conditional Random Fields (CCRF) [63], which has been employed for structured regression for AU intensity estimation [8]. Further extensions, which combine ANNs with CRF and CCRF models respectively, are Conditional Neural Fieldss (CNFs) and Continuous Conditional Neural Fieldss (CCNFs). In these models, each hidden component in this model consists of a neutral layer. These methods have been employed for expression recognition [81] and AU intensity estimation [8].

Alternative methods employed for temporal modelling, other than graphical models. These include manifold learning techniques used to embed image sequences as lines that can be traced through a low dimensional representation of the expression space [21]. Gaussian Processes (GPs) are stochastic processes which consist of a set of random variables, any finite number of which have Gaussian distributions. They are generally designed for use with time-series data, and have been employed for facial expression classification [22, 23].

2.8 Databases

During the past two decades a number of facial expression databases have been created in order to be used for expression and AUs analysis. In this section the existing 2D and 3D databases are reviewed, including both posed and spontaneous examples of expressions, and include details of their content and size, types of data, and the annotation that is useful for researchers working in this field.
2. Review of Facial Expression Analysis

2.8.1 2D Databases

Numerous 2D facial expression databases have been created, and the data that they contain varies widely in terms of expressions, number of subjects, and annotations. Here an overview of the most useful 2D datasets available for facial expression analysis, that have been employed for training and testing of systems in the last 10 years, are included. They are also summarised in the top half of Table 2.1.

Posed expressions initially formed the basis for expression recognition system testing, and there are a number of posed expression 2D databases that are still widely used in facial expression analysis validation, due to the quality and variety of data they contain. These include the MMI database [118], which consists of 19 subjects performing a very wide range of AU combinations, and is fully FACS coded with AU segments, and the Cohn-Kanade (CK) database [72], which has a larger number of subjects, 97, performing a smaller number of combinations. The latter database is also FACS coded with the full set of AUs, but not with segments. Another two databases, the ISL Frontal [172] and ISL Multi-view [171], are available with 10 and 8 subjects respectively, coded with up to 15 AUs and for which there are 34 facial points given. Finally, the GEMEP-FERA database [9] aims to provide data which is more realistic, though still posed, as it contains 10 actors who were not given instructions on how to perform the requested expressions. This data includes 18 different emotional states, and is FACS coded with 12 AUs including intensities.

As it has been shown that posed expressions vary greatly from those displayed during real emotion [210], more recent databases have aimed to capture spontaneous expressions elicited in a number of ways. The Sensitive Artificial Listener (SAL) database [44] engaged 20 subjects in conversation with an avatar in order to elicit spontaneous emotions and expressions. The data was coded for emotion intensity, valence, activation and masking. This work then lead to the SEMAINE database [100], which used the same procedure to elicit a range of emotions from 150 subjects. This database was annotated with 5 affective dimensions, as well as 27 other categories. Selected extracts from this database were also FACS coded. In order to provide spontaneous expression data, the groups behind the CK database extended their data into the Extended Cohn-Kanade (CK+)
2.8. Databases

database [93]. This contained annotated sequences taken from the parts in between the previous recordings, when the subject interacted with the experimenter, generally by smiling, laughing or talking. Suitable examples were selected and fully FACS coded for inclusion in the new database. In addition, AAM points were provided for all images in this new database. Two more very recently released databases are the Spontaneous Micro-expression Database (SMIC) [121] which contains 164 examples of micro-expressions from 20 subjects with emotion labels, and the Chinese Academy of Sciences Micro-expression (CASME) database [198] which contains examples of 35 subjects performing spontaneous expressions in reaction to stimuli videos. In both of these databases the subjects were instructed to remain neutral during the process, hence the expressions that result are micro-expressions (brief flashes of expressions that last only a fraction of a second). The sequences in the latter database have also been FACS coded with the segments of the AUs (onset, apex and offset).

Finally, there are two databases that provide not only spontaneous expression examples, but also AU intensity coding. Providing data of this kind is a highly time-consuming task, which is why it only recently that databases have begun to include these codings. The first of these, the UNBC-McMaster Shoulder Pain Expression Archive database [94], focussed on collecting pain expressions. This was done by recording 129 subjects who were suffering with a shoulder pain while they performed some routine exercises on their shoulder. This data was FACS coded with intensities for 11 AUs, and 66 facial points were also provided. The most recent database to provide intensity coded spontaneous data is the Denver Intensity of Spontaneous Facial Action (DISFA) database [98]. This consists of 27 subjects recorded whilst watching a series of videos designed to elicit a wide range of emotions. This database has been annotated with 12 AUs, and also provides 68 facial points.

2.8.2 3D Databases

Almost all of the 3D facial expression databases collected thus far contain only posed examples of expressions, due to the difficulty of eliciting spontaneous emotions under the restrictive 3D recording setups. Systems that have been employed for acquisition of
## 2. Review of Facial Expression Analysis

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>S/D</th>
<th>P/S</th>
<th>Size</th>
<th>Content</th>
<th>Landmarks</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CASME [198]</td>
<td>D</td>
<td>S</td>
<td>35 adults</td>
<td>micro-expressions: H, D, F, Sa, A</td>
<td>N/A</td>
<td>AU’s with segments and emotion labels</td>
</tr>
<tr>
<td></td>
<td>CK [72]</td>
<td>D</td>
<td>P</td>
<td>97 adults</td>
<td>23 AU combinations</td>
<td>N/A</td>
<td>43 AUs</td>
</tr>
<tr>
<td></td>
<td>CK+ [93]</td>
<td>D</td>
<td>S</td>
<td>66 adults</td>
<td>smiles and speech</td>
<td>68 facial points</td>
<td>8 AUs</td>
</tr>
<tr>
<td></td>
<td>DISFA [59]</td>
<td>D</td>
<td>S</td>
<td>27 adults</td>
<td>spontaneous H, Su, D, Sa</td>
<td>68 facial points</td>
<td>12 AUs with intensities</td>
</tr>
<tr>
<td></td>
<td>GEMEP-FERA [9]</td>
<td>D</td>
<td>P</td>
<td>10 adult actors</td>
<td>range of 18 emotions inc. six basic, C, amusement, relief</td>
<td>N/A</td>
<td>emotions ratings and 12 AUs</td>
</tr>
<tr>
<td>2D</td>
<td>ISL Frontal [172]</td>
<td>D</td>
<td>P</td>
<td>10 adults</td>
<td>Various</td>
<td>N/A</td>
<td>14 AUs</td>
</tr>
<tr>
<td></td>
<td>ISL Multi [171]</td>
<td>D</td>
<td>P</td>
<td>8 adults</td>
<td>Various</td>
<td>34 facial points</td>
<td>15 AUs</td>
</tr>
<tr>
<td></td>
<td>MMI [118]</td>
<td>D</td>
<td>P</td>
<td>19 adults</td>
<td>79 AU combinations</td>
<td>N/A</td>
<td>AU’s with segments</td>
</tr>
<tr>
<td></td>
<td>SAL [44]</td>
<td>D</td>
<td>S</td>
<td>20 adults</td>
<td>Various</td>
<td>N/A</td>
<td>Emotion intensity, valence, activation and masking</td>
</tr>
<tr>
<td></td>
<td>SEMAINE [100]</td>
<td>D</td>
<td>S</td>
<td>150 adults</td>
<td>Various</td>
<td>N/A</td>
<td>5 dims, 27 cats, some AUs</td>
</tr>
<tr>
<td></td>
<td>SMIC [121]</td>
<td>D</td>
<td>S</td>
<td>20 adults</td>
<td>micro-expressions</td>
<td>N/A</td>
<td>Emotion labels</td>
</tr>
<tr>
<td></td>
<td>UNBC-McM [94]</td>
<td>D</td>
<td>S</td>
<td>129 adults</td>
<td>pain expressions</td>
<td>66 facial points</td>
<td>11 AUs with intensities</td>
</tr>
<tr>
<td>3D</td>
<td>Bosphorus [138]</td>
<td>S</td>
<td>P</td>
<td>105 adults inc. 27 actors</td>
<td>24 AUs, neutral, 6 basic exps, occlusions</td>
<td>24 facial points</td>
<td>25 AUs with intensities</td>
</tr>
<tr>
<td></td>
<td>BU-3DFE [207]</td>
<td>S</td>
<td>P</td>
<td>100 adults</td>
<td>6 basic exps at 4 intensity levels</td>
<td>83 facial points</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>BU-4DFE [204]</td>
<td>D</td>
<td>P</td>
<td>101 adults</td>
<td>6 basic exps</td>
<td>83 facial points</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>BP4D-Spon [215]</td>
<td>D</td>
<td>S</td>
<td>41 adults</td>
<td>reaction to 8 tasks: aimed for H, Sa, Su, E, F, P, A, D</td>
<td>83 facial points</td>
<td>27 AUs with intensities</td>
</tr>
<tr>
<td></td>
<td>D3DFACS [35]</td>
<td>D</td>
<td>P</td>
<td>10 adults inc. 4 FACS experts</td>
<td>Up to 38 AUs per subject</td>
<td>N/A</td>
<td>AU peaks</td>
</tr>
<tr>
<td></td>
<td>ICT-3DRFE [159]</td>
<td>S</td>
<td>P</td>
<td>23 adults</td>
<td>15 exprs: inc. basic, neutral</td>
<td>N/A</td>
<td>AU’s with intensity levels</td>
</tr>
</tbody>
</table>

Table 2.1: Publicly available facial expression databases. S/D: Static or dynamic data. P/S: Posed only or contains spontaneous data. Size: Number of subjects. Content: Expressions or AUs available (H/Sa/A/D/Su/E/P/C/L - Happy/Sad/Angry/Disgust/ Surprise/Embar- rassment/Pain/Contempt/Laugh). Landmarks: Available landmarks. Annotation: Available annotation.
2.8. Databases

Figure 2.8: Examples from some of the facial expression databases currently publicly available. (a) CK+ database (b) DISFA database (c) The Bosphorus database (d) D3DFACS (e) BU-4DFE database (f) BP4D-Spontaneous.
these databases include the static DI3D and dynamic DI4D systems [92] which employ six cameras, three on each side of the subject. Four are exploited for stereo reconstruction of the facial geometry, and two for capturing the colour, and the 3DMD system [1], which employs a similar set up. Other methods that have been employed for database acquisition are structural light reconstruction, where a known pattern of infra-red lights are projected onto the subject [14], and photometric stereo, where lights are flashed in quick succession from a number of different angles in order to observe different lighting and shadows for the purpose of reconstruction [62].

The first systematic effort to collect 3D facial data for facial expression recognition resulted in the creation of BU-3DFE dataset [207]. Static 3D expressive faces of 100 subjects were captured with the DI3D system, displaying the six prototypical expressions at four different intensity levels. The models created were of resolution in the range of 20,000 polygons to 35,000 polygons, depending on the size of subject’s face. The database was accompanied by a set of metadata including the position of 83 facial feature points on each facial model, as depicted in Fig. 2.9. The same institution continued the effort and recorded BU-4DFE [204], the first database consisting of 4D faces (sequences of 3D faces) captured with the DI4D system. The database includes 101 subjects and contains sequences of the six prototypical facial expressions with their temporal segments (offset, onset and apex) with each sequence lasting approximately 4 secs (examples can be seen in Fig. 2.8e). The temporal and spatial resolution are 25 frames/sec and 35,000 vertices, respectively. Unfortunately, the database provides no AUs annotation.

Another publicly available dataset consisting of static 3D facial models is the Bosphorus database [139]. The database consists of 105 subjects (60 men and 45 women, with the majority of the subjects being Caucasian), 27 of whom were professional actors/actresses, in various poses, expressions and occlusion conditions. The subjects expressed the 6 six prototypical facial expressions (examples can be seen in Fig. 2.8c), and up to 24 AUs. The database was captured using a structured light method, and is fully annotated with regards to 25 AUs, split as lower (18) AUs and upper (7) AUs. The texture images are of resolution 1600 × 1200 pixels while the 3D faces consist of approximately 35,000 vertices. The database is accompanied by a set of available metadata consisting of 24 manually
labeled facial landmarks such as nose tip, inner eye corners, etc.

More recently the ICT-3DRFE database [160] was publicly released. This database consists of 3D data of very high resolution recorded under varying illumination conditions using a photometric stereo method, in order to test the performance of automatic facial expression recognition systems. The database contains 3D models for 23 subjects (17 male and 6 female) and 15 expressions: the six prototypical expressions, two neutral states (eyes closed and open), two eyebrow expressions, scrunched face expression, and four eye gaze expressions. Each model in the dataset contains up to 1,200,000 vertices with reflectance maps of $1296 \times 1944$ pixels, resolution that corresponds to a detail level of sub-millimeter skin pores. The ability to relight the data is ensured by the reflectance information provided with every 3D model. This information allows the faces to be rendered realistically under any given illumination. The database also includes photometric information that allows photorealistic rendering. The database is fully annotated with regards to AUs. AUs are also assigned scores between 0-1 depending on the degree of muscle activity.

The first database to contain coded examples of dynamic 3D AUs, namely the D3DFACS, was presented in [35]. It contains 10 subjects, including 4 FACS experts, performing posed examples of up to 38 AUs, and is annotated with a single AU code for each sequence. The database was acquired with the 3DMD system. In total, 519 AUs sequences were captured at 60 frames/sec, consisting of approximately 90 frames each. The peak of each sequence has been coded by a FACS expert. It was the first database to allow research into dynamic 3D AU recognition and analysis.

Finally, very recently a new 3D database has been released which contains the first examples of spontaneous expressions recorded in 3D, using the DI4D system. This is the BP4D-Spontaneous database [215], and has been recorded by the same lab as the BU-3DFE and BU-4DFE databases. The database consists of 41 adults recorded while reacting to 8 tasks designed to elicit amusement, embarrassment, pain and 5 of the basic expressions. It has been annotated with 27 AUs, and includes the 83 facial points included in the previous databases. This database is a very important development in the field of 3D facial analysis, as it will allow systems to be trained and tested on real displays of
2. Review of Facial Expression Analysis

(a)

(b)

Figure 2.9: Examples of the facial points included in facial expression databases (a) DISFA database (b) BU-3DFE and BP4D-Spontaneous databases.

emotion for the first time.

A number of researchers have begun exploring the use of the Kinect device for acquisition of 3D facial expression data. However, the current device is no yet high enough resolution, and is not of the required accuracy, for capturing the subtleties of facial motion, beyond obvious and exaggerated expressions. However, it is expected that soon devices of these kind will be adequate for capturing expressions, and so will make acquisition of databases of these kind easier and cheaper, as well as allowing for less intrusive systems that reduce the natural behaviour possible. This would also allow for collection of 3D data in a much wider range of situations, which would allow 3D facial expression to become a possibility in real-life applications.
2.9 Performance Measures

In this section an overview of the main measures of performance that are typically used in facial analysis systems, and will be employed in this thesis, are detailed. This includes measures for both classification and regression.

2.9.1 Classification Performance

Classifiers can be either binary, separating data into two classes (positive and negative), or multiclass, separating the examples into several classes. In the majority of cases, similar performance measures can be employed to both types, with slight variations: recall rate, precision, F measures. However, only in the binary case can a final measure, the Receiver Operating Characteristic (ROC) curve, be employed.

Recall Rate The recall rate is the simplest way to measure performance of a classifier, and is sometimes referred to as the classification rate. For a binary classifier, it is calculated by taking the number of true positives, and dividing it by the total number of positive examples:

$$Recall = \frac{TP}{TP + FN}$$

(2.3)

where $TP$ is the number of true positives (positive examples that were classified as positive), and $FN$ is the number of false negatives (positive examples that were classified negative).

For a multiclass classifier it can be similarly calculated for each class:

$$Recall_C = \frac{TC}{TC + FC}$$

(2.4)

where $TC$ is the number of true classifications for class $C$, and $FC$ is the number of false classifications of class $C$ examples as other classes.

This performance measure is used very widely in the validation of facial expression analysis systems. However, it does not always give an accurate indication of the discriminative ability of the system, as it is possible for a high rate to be given when the classifier even when the classifier gives no useful information about the data. For example, if the result of the system is 100% positive classification regardless of input, then the recall rate will be 100%, as all positive examples were correctly classified, despite the fact that this is clearly
not a useful classifier. For this reason, other more sophisticated measures of performance are becoming more widely used in classification problems.

**F$_1$ Measure** In order to calculate a more accurate indication of overall performance of the system, it is necessary to combine two metrics. Firstly recall, but also the precision rate, which measures the percentage of the positively classified examples that are true positives:

$$
Precision = \frac{TP}{TP + FP}
$$

(2.5)

where $FP$ is the number of false positives (negative examples that were classified as positive). Again, it can similarly be defined in the multiclass case.

These two metrics can be combined using an F$_\beta$ measure [182]:

$$
F_\beta = \frac{(1 + \beta^2)^\frac{\text{Precision} \times \text{Recall}}{\beta^2 (\text{Precision} + \text{Recall})}}
$$

(2.6)

A special case of this metric, F$_1$, is usually employed. In this case the recall and precision rates are equally balanced:

$$
F_1 = 2 \left( \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right)
$$

(2.7)

**Confusion Matrix** In order to display the results of a classification task (usually in the multiclass case), a confusion matrix is often used. In this matrix, each row represents the true class labels, and the columns represent the predicted class labels given by the classifier output. If the classifier is 100% accurate, then the only non-zero entries in the matrix will be on the diagonal. However, non-diagonal entries indicate mistakes made by the classifier, hence the level of its “confusion”.

Table 2.2 shows an example confusion matrix for a 3-class set of results. This example also shows how the matrix can be used to calculate the recall and precision rates directly, simply by dividing the diagonal values by the sums of the rows and columns respectively. This then allows the F$_1$ score to be calculated, as is shown in the final row. This example demonstrates how the F$_1$ can be substantially higher or lower than recall alone, hence giving a more accurate representation of the performance in each class.
2.9. Performance Measures

<table>
<thead>
<tr>
<th></th>
<th>Predicted</th>
<th></th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 10 3 2</td>
<td></td>
<td>0.67</td>
</tr>
<tr>
<td>2</td>
<td>1 11 3</td>
<td></td>
<td>0.73</td>
</tr>
<tr>
<td>3</td>
<td>4 3 8</td>
<td></td>
<td>0.53</td>
</tr>
<tr>
<td>Precision</td>
<td>0.67 0.65 0.61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1 Score</td>
<td>0.67 0.69 0.57</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: Example confusion matrix

Receiver Operator Characteristic Curve The $F_1$ score gives a snapshot of the performance of a binary classifier, but as class decisions often depend on a dividing line or threshold, that can be varied, it doesn’t necessarily give the highest potential performance of that classifier. A better way to measure this is the ROC. This metric uses confidence values about each class, and varies the threshold of the decision in order to plot the full performance.

The ROC curve [220] shows how the true positive rate and false positive rate vary with respect to each other as this threshold is altered. At one extreme the true and false positive rates will both be zero, when all examples are on negative side of the threshold, and at the other extreme, the true and false positives will both be 1, when all of the positive and negative examples have been classified as positive. In between, the curve will stay above the diagonal joining these two extremes. An example of this can be seen in Fig. 2.10.

In order to calculate a single score for the performance of the classifier, the Receiver Operating Characteristic Area Under the Curve (ROC AuC) is usually used. As this measure is only applied in the binary classifier case, and thus the confidence values can always be reversed, this score will always been higher than or equal to 50%.

2.9.2 Regression Performance

Performance measures for regression aim to give an indication of how closely the regressor can match the input labels. A number of different measures are typically employed for
2. Review of Facial Expression Analysis

Figure 2.10: Example ROC curve

this purpose: the Root Mean Squared Error (RMSE), Pearson Correlation Coefficient (PCC) and Intraclass Correlation Coefficient (ICC).

Root Mean Squared Error  The RMSE is simply a measure of the mean error introduced by the regressor predictions from the true labels. It is calculated as:

\[
RMSE(\hat{Y}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{Y}_i - Y_i)^2}
\]  

(2.8)

where \(\hat{Y}\) are the predicted labels and \(Y\) are the true labels.

It is able to give an indication of how closely, on average, the predicted labels follow the true labels. However, for a very noisy prediction it the score may not give a clear representation of how well the regressor is performing, because it cannot show whether
there is a common pattern between the true and predicted labels, only how far apart they are on average. For this reason other measures are generally required.

**Pearson Correlation Coefficient** In order to measure more accurately how closely the true and predicted labels follow a similar trend, it is useful to measure the correlation between the two sets of labels. One measure of this that is commonly used is the PCC. The definition is as follows:

\[
PCC = \frac{\sigma_{\hat{Y},Y}}{\sigma_{\hat{Y}} \sigma_{Y}}
\]

(2.9)

where \(\sigma_{\hat{Y},Y}\) is the covariance between the two sets of labels, and \(\sigma_{\hat{Y}}\) and \(\sigma_{Y}\) are the standard deviations of the predicted and true labels respectively.

Although this metric produces a good measure of how well the variations in two signals are correlated, it does not measure absolute difference between the two signals. This means that the actual predictions could be very different from the ground truth, but still give a high PCC measure.

**Intraclass Correlation Coefficient** An alternative correlation measure, the ICC [150], has also been used in a number of works on intensity estimation. This measure, which is most commonly used for measuring the agreement of two or more annotators, has the benefit of taking into account the absolute difference between the sets of labels, as well as whether they follow the same general trend, whereas the PCC only measures the latter. For this reason, the ICC is becoming more popular as a measure of regressor performance.

Generally, the ICC allows one-way correlations to be measured, as it performs on groups of data, thus allowing the targets within groups to be ordered randomly. However, when measuring regression performance it is known that the ground truth and predicted values are being compared, so it can be taken that each set of data comes from a single distinct ‘judge’. Therefore, for \(k\) judges, if \(\sigma_{T}^{2}\) is the variance of the targets, \(\sigma_{E}^{2}\) is the variance of the assumed error in the judge’s scoring, and \(\sigma_{I}^{2}\) is the variance of an interaction component that determines how the judge’s decisions are affected by particular targets:

\[
BMS = k\sigma_{T}^{2} + \sigma_{E}^{2}
\]

(2.10)
2. Review of Facial Expression Analysis

is the between-targets mean square and

\[ EMS = \frac{k\sigma_i^2}{k-1} + \sigma_E^2 \]  

(2.11)

is the residual sum of squares due to the assumed error.

The ICC employed then takes the following form:

\[ ICC = \frac{BMS - EMS}{BMS + (k-1)EMS} \]  

(2.12)

2.10 Summary and Challenges

This review has demonstrated how there are some aspects of research into facial expression analysis which are maturing, with a large number of works proposing successful methods for expression and AU recognition in many of the databases available. However, the ability to fully analyse and understand the emotional state of the subject is still far from a solved problem, with many parts of this task little explored. There are still many issues to overcome before robust expression analysis systems can be used in real-world applications.

In this section, a number of areas in which little or no work has yet been conducted is summarised. Three of these areas: 3D features, 3D dynamic analysis, and AU intensity estimation, motivate the remaining chapters of this thesis. The final area, 3D spontaneous expression analysis, is as yet an open area of research which will focus future work.

2.10.1 3D Features

Though a large amount of research has been conducted into possible 3D features suitable for facial expression analysis, the vast majority of these have employed facial feature points and their movement. However, facial feature points are unable to capture the entire range of movements possible in the face, and there are a number of AUs which, when displayed, do not affect facial points in a distinctive manner. For example, AU 6 (Cheek Raiser), AU 9 (Nose Wrinkler) and AUs 23 and 24 (Lip Tightener vs. Lip Presser) all affect textures on the surface of the face, without altering feature points significantly. For this reason, more research is needed into features that exploit the full face - holistic features.
Binary pattern features are one such family that have been widely exploited for facial expression analysis of 2D data, and yet little research as thus far been conducted into their use for analysis of 3D facial geometries. Hence, methods for applying and extending these features to the 3D case, and the benefits that this can bring for the problem of AU detection, could be explored.

2.10.2 3D Dynamics

Until recently, databases available for use for 3D facial expression analysis contained static data only. As a result, the majority of research this far has been conducted on data of this kind. However, as 4D dynamic video data has become available, the trend has started to shift the interest of the researchers towards the analysis of facial expression dynamics in 3D data, which allows the encoding of temporal cues that are indicative of more complex states and expressions. Currently, there exist three publicly available datasets of dynamic 3D facial samples, namely BU-4DFE, D3DFACS and the recently released BP4D-Spontaneous, with only the latter two containing FACS codings. As more databases containing data of this nature are collected, new systems are required to perform accurate recognition across different databases.

2.10.3 AU Intensities

One area of research that has not been tackled in detail thus far is that of AU intensity estimation. The main reason for the limited work done in this area the fact that until recently, there was little data available that had been FACS coded with full AU intensities. However, due to the UNBC-McMaster pain database, and the recent release of the DISFA database, the interest in this area is increasing greatly. This is further enhanced by the fact that both of these databases contain spontaneous data. As the dynamics of posed expressions have been shown to significantly differ from those of real, spontaneous, expressions [178], this is particularly important for allowing work that aims to explore true interactions and correlations between different AUs.

Until very recently, all of the databases that contained FACS intensity coding consisted of 2D data only. For this reason it has not been possible to investigate AU intensity
estimation in 3D data, and hence in this work 2D data is employed. However, it is expected that similar methods could be employed on 3D data with the appropriate features, on the new 3D data that has become available.

2.10.4 3D Spontaneous Data

Research in posed 2D expression and AU recognition, particularly in frontal data (as available in the majority of databases) is now quite mature. A large number of techniques have been developed for tackling this problem, and the performance scores achieved in these databases by the state-of-the-art is very high. Spontaneous data, as well as being much more difficult to capture, presents more of a challenge for analysis systems, as the range and display of expressions show more variation, and the subject is more likely to move their head, speak, or perform simultaneous gestures. However, a number of works have managed to achieve good recognition performance of AUs in databases containing this kind of data [11].

As the BP4D-Spontaneous data was only released last year, and thus far is the only database in existence that contains spontaneous examples of 4D expressions, existing works in the field of facial expressions in 3D are all based on databases of acted, exaggerated expressions of the six basic emotions, although they rarely occur in our daily life. Yet, increasing evidence suggests that deliberate or acted behaviours differ in appearance and timing from spontaneous ones [3]. For instance, acted smiles have larger amplitude, shorter duration, and faster onset and offset velocity than naturally occurring smiles [179]. In turn, automatic approaches trained in laboratory settings on recordings of acted behaviour fail to generalise to the complexity of expressive behaviours found in real-world settings. For this reason, future work in 3D facial expression analysis will likely shift towards analysis of this kind of data.
Binary Pattern Features for 3D Facial Action Unit Detection

The preliminary results for this work were published in [SZP12b] and [SZP12a].

3.1 Introduction

LBPs [109] is a family of techniques that has been widely applied to perform feature extraction for the problems of facial expression, as described extensively in Section 2.3.1. Due to their success in 2D facial expression applications, in this chapter, the application of these techniques to the problem of 3D facial geometry analysis is explored. The aim of this chapter is to explore a number of binary pattern variants, that have been successfully applied to 2D intensity images, for use on 3D facial geometry data.

A set of binary pattern based features are proposed to fully explore the benefits of applying binary pattern operators to the problem of 3D facial AU detection. Two methods are proposed for extracting LBPs from normals, as well as two spherical representations of the 3D data, the Azimuthal Projection Distance Image (APDI) and Azimuthal Projection Component Images (APCIs), and the application of binary patterns to each of these. Gabor and Monogenic filtering techniques are also applied, along with the LPQs operator, to both the depth map and spherical representations. These features are based on several different representations of the 3D facial information, and exploit ideas from 2D methods.
3. Binary Pattern Features for 3D Facial Action Unit Detection

that have been previously proposed.

A framework for testing the relative performance of these features is presented. Feature vectors are formed for each feature type through concatenation of histograms formed from the resulting binary numbers. Feature selection is then performed using a novel two-stage GB approach, which first selects regions, and then individual features. Finally, SVMs are applied as classifiers for detection of each AU.

Experiments are conducted to perform parameter optimisation, explore the benefits of the region selection step, and assess the individual feature performance, through 10-fold cross-validation testing. Their effectiveness is compared to that of the original 2D techniques and previously proposed 3D feature extraction methods, and explore the benefits of combining multiple modalities through fusion of the 2D and 3D features of different kinds. Low AU intensity and cross-database experiments are also conducted to further explore the performance of the features.

The contributions of this chapter are as follows:

- The introduction of novel histogram-based descriptors, inspired by LBP s and LPQs, of various facial surface representations. In particular, this idea is applied to create two categories of features:
  - Basic features: Representations based on the facial mesh surface normals are employed, both directly and via a measure of curvature through use of the APDI and APCIs.
  - Filter-based features: Filtered and transformed depth maps and APDIs are employed, through use of phase quantisation, Gabor and Monogenic techniques.
- A novel method for selection of regions in the image, based on the GB algorithm, prior to selection of individual features.
- A novel classification level fusion method that is shown to outperform feature level fusion.
3.2 Facial Geometry Representations

- A wide variety of experiments to test the proposed features, including cross-database testing.

This chapter is organised as follows: Section 3.2 details the representations exploited by the employed features, Section 3.3 then outlines the filtering techniques applied to these representations as a preprocessing step to allow application of the binary pattern methods. The necessary adaptations of the traditional techniques are explained in Section 3.4. In Section 3.5, the methodology for testing and comparison of the feature performance is outlined. The experiments conducted are described in Section 3.6, and finally Section 3.7 concludes the chapter.

3.2 Facial Geometry Representations

The exploitation of four different representations of the 3D facial geometry are examined: the widely used depth map representation, the facial geometry normals, and two different representations based on the Azimuthal Equidistant Projection (AEP) - the APDI and the APCIs - which are representations of normals in Euclidean space.

3.2.1 Depth Map

The depth map representation is widely used in 3D facial analysis (e.g. [12, 185, 60]) as it is a very simple 2D representation. In this work, 2D Delaunay triangulation is performed in this $x$-$y$ plane on the facial mesh points. This triangulation is then used to interpolate the $z$ values onto a regular grid of defined $x$ and $y$ values with resolution of 300x300 to form the depth map. An example of this can be seen in Fig. 3.1b for the facial mesh in Fig. 3.1a.

3.2.2 Facial Mesh Normals

The normals of the facial geometry data are employed directly to create regular grids in a similar way to the depth map. Three regular $x$-$y$ grids are created, one for each component of the normals. Then the values are interpolated using Delaunay triangulation across
3. Binary Pattern Features for 3D Facial Action Unit Detection

Figure 3.1: 2D representations of the facial mesh for subject bs043 from the Bosphorus database performing AU20 (a) Original facial mesh (b) Interpolated depth map (c) Facial mesh normals (d) Azimuthal Projection Distance Image (e)-(f) Azimuthal Projection Component Images

these grids, and the resulting vectors normalised. These grids are then used directly for calculation of the normal-based feature types, and also for deriving the APDI and APCIs.

3.2.3 Azimuthal Projection Images

Since normals do not lie in a Euclidean space, they must be projected (or ‘unwrapped’) into this space before the curvature differences can be measured. The AEP has previously been employed to capture the local variations in facial shape [152], and can be applied to the face normals, in order to project each 3D direction onto the position in a Euclidean 2D plane. So for a regular grid of normals, defined as \( \mathbf{n}(i, j) = (u_{i,j}, v_{i,j}, w_{i,j}) \), the AEP
point \( p(i, j) = (x_{i,j}, y_{i,j}) \) in this plane is defined as:

\[
\begin{align*}
x_{i,j} &= k' \cos(\theta(i, j)) \sin(\phi(i, j) - \hat{\phi}(i, j)) \\
y_{i,j} &= k' \cos(\hat{\theta}(i, j)) \sin(\phi(i, j)) \\
&\quad - k' \sin(\hat{\theta}(i, j)) \cos(\phi(i, j)) \cos(\phi(i, j) - \hat{\phi}(i, j))
\end{align*}
\] (3.1)

where \( \theta(i, j) = \frac{\pi}{2} - \arcsin(w_{i,j}) \) is the elevation angle measured from the z-axis, \( \phi(i, j) = \arctan\left(\frac{v_{i,j}}{u_{i,j}}\right) \) is the azimuth angle, \( \hat{\theta}(i, j) \) and \( \hat{\phi}(i, j) \) are the elevation and azimuth of the mean normal \( \hat{n}(i, j) \) at the point \( p \), \( k' = \frac{c}{\sin(c)} \) and

\[
\cos(c) = \sin(\hat{\theta}(i, j)) \sin(\theta(i, j)) + \cos(\hat{\theta}(i, j)) \cos(\theta(i, j)) \cos(\phi(i, j) - \hat{\phi}(i, j))
\] (3.2)

However, for the purpose here, it is necessary to be able to directly compare the projection coordinates of neighbouring points. Because the mean normal would be different at every point in the mesh, if it were employed as above, this would now allow a comparison to be made between the value at different points in a particular mesh. As binary pattern features rely on thresholding, and thus a fair comparison between points, this would cause problems. Hence, instead of employing the true mean normal, \( \hat{\theta} \) and \( \hat{\phi} \) at every point are set to be \( \frac{\pi}{2} \) and 0 respectively, so that the distance calculated is always compared to a normal \( \hat{n} = (1, 0, 0) \) which was chosen as a reference to create an image suitable for further analysis.

This assumption makes \( \cos(c) = \sin(\theta(i, j)) \) and allows the projection to be simplified to:

\[
\begin{align*}
x_{i,j} &= k' \cos(\theta(i, j)) \sin(\phi(i, j)) \\
y_{i,j} &= k' \cos(\theta(i, j)) \cos(\phi(i, j))
\end{align*}
\] (3.3)

The above formulation then allows distances between the normals in Euclidean space to be directly found, and this simplification also reduces the complexity of the feature extraction process. In order to employ this in the binary pattern framework, images are formed from the projection in two ways. First, this formulation is used to extract a measure of
3. Binary Pattern Features for 3D Facial Action Unit Detection

![Facial Mesh Normals](image)

Figure 3.2: Facial Mesh Normals for subject bs043 performing AU20 (a) normals in the whole mesh (b) zoomed image normals

curvature which is taken as an absolute distance from the origin $d_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}$, and these values form the APDI for the facial mesh. An example of this calculated for the facial mesh seen in Fig. 3.1a can be seen in Fig. 3.1d. Secondly, each component of the projection is taken, $x$ and $y$, as APCIs. Examples of these can be seen in Figs. 3.1e and 3.1f.

3.3 Preprocessing of Facial Representations

A number of the features proposed require a preprocessing step, prior to the application of the binary operation. At this stage, two filtering techniques, Gabor and Monogenic, are employed for multiscale analysis in order to capture features of the facial geometry representations, as described here.

3.3.1 Gabor Filters

Gabor filters have been widely used 2D facial expression recognition [172, 194], and also applied to 3D analysis [143], as they are well suited to capturing the structural information in an image at different orientations and scales in a way that is similar to the human visual system [50]. The frequency domain transfer function of the filters used consists of a radial
log Gabor filter multiplied by an angular Gaussian component:

\[ G(\mathbf{u},\nu,\theta) = \exp \left( -\frac{(\log(\nu|\mathbf{u}|))^2}{2(\log \sigma)^2} \right) \exp \left( -\frac{(\angle \mathbf{u} - \theta)^2}{2\sigma^2} \right) \]  (3.4)

where \( \nu \) and \( \theta \) are the scale and orientation of the filter respectively, and \( \sigma \) and \( \sigma_\phi \) define the spread of the filter in the radial and angular directions respectively. The Fourier transform of the image is multiplied by this function at four different scales and four orientations, and then the inverse transform taken to find the resulting Gabor coefficients, \( g(x,y) = F^{-1}(GF(I)) \). The magnitude and phase of these are then taken as new images, \( g_M = |g(x,y)| \) and \( g_P = \angle g(x,y) \). Examples of the magnitude and phase images, produced from the depth map and APDI representations, can be seen in Figs. 3.5a-3.5b and 3.5c-3.5d respectively.

### 3.3.2 Monogenic Filters

The monogenic signal is an alternative approach for image analysis. It is the 2D equivalent of the analytic signal, which extracts the local amplitude and phase information from the 1D signal through use of the Hilbert transform. This applies a \( \pi/2 \) transform in the Fourier transform, thereby creating a quadrature pair of filters which allows the phase to be calculated. The monogenic signal uses the Riesz transform to extend this idea to 2D signals, and extracts a 2D representation of the phase, which gives enough degrees of freedom to capture the full structural information about the image. This results in three outputs - magnitude, which captures the quantitative local amplitude information, and phase and orientation, the two components of the phase measurement. The advantage of this method is that it is no longer necessary to apply filters at multiple orientations, as with Gabor filters, as features in every direction are captured simultaneously. This greatly reduces the size of the feature descriptors, and therefore the computation demand. However, multiple scales are still useful for capturing different structural resolutions in the image.

In practice, the three components of the monogenic signal, magnitude \( m_M \), phase \( m_P \) and orientation \( m_O \), can be calculated through the use of two orthogonal monogenic filters.
with transfer functions $H_1(u) = \frac{iu_1}{|u|}$ and $H_2(u) = \frac{iu_2}{|u|}$, and radial log Gabor filters with varying scales:

$$m_M = \sqrt{g'(x, y)^2 + h'_1(x, y)^2 + h'_2(x, y)^2}$$

$$m_P = \arctan \left( \frac{h'_2(x, y)}{h'_1(x, y)} \right)$$

$$m_O = \arctan \left( \frac{g'(x, y)}{\sqrt{h'_1^2 + h'_2^2}} \right)$$  \hspace{1cm} (3.5)

where $h'_i = F^{-1}(H_i G' F(I))$, $I$ is the image, $F$ is the 2D Fourier transform, and $G'(u)_\nu = \exp \left( - \frac{\log(|\nu|)^2}{2 \log(\sigma)^2} \right)$ is the transfer function of the radial component of the full log Gabor filter outlined in the Section 3.3.1 and $g' = F^{-1}(G' F(I))$. Examples of the magnitude, phase and orientation images, produced from both the depth map and APDI representations, can be seen in Figs. 3.5i-3.5k and 3.5l-3.5n respectively.

### 3.4 Binary Pattern Features

In this section the proposed set of new binary pattern features for analysis of 3D facial geometry information for AU detection is explained in detail. Firstly the original LBP feature is described, and its extension using normals, and the APDI and APCIs. Then filter-based methods, both utilising the depth map and APDI, are described. The full set of proposed feature types is as follows.

**Basic features:**

1. Local Azimuthal Binary Patterns (LABPs).
2. Local Normal Binary Patterns (LNBPs).

**Filter-based features:**

1. Local Depth Phase Quantisers (LDPQs).
2. Local Azimuthal Phase Quantisers (LAPQs).
3. Local Depth Gabor Binary Patterns (LDGBPs).
4. Local Azimuthal Gabor Binary Patterns (LAGBPs).
3.4. Binary Pattern Features

5. Local Depth Monogenic Binary Patterns (LDMBPs).

6. Local Azimuthal Monogenic Binary Patterns (LAMBPs).

3.4.1 Local Depth Binary Patterns

Local Depth Binary Patterns (LDBPs) were applied to the problem of facial recognition of 3D meshes in [61]. They exploit a 2D representation of the 3D information, the depth map interpolated onto a regular grid, in order to encode the local shape around each point in the mesh. This operation is shown in Figs. 3.3a-3.3c. A circular neighbourhood is defined around each pixel with \( P \) points regularly spaced around a circle of radius \( r \). The central pixel value is then used as a threshold to assign binary bits to the pixels in the neighbourhood, thus producing a binary number for that pixel in the following way. Let \( I^D \) be the depth map image. Then the LDBP operator for a central point \( I^D(x_c, y_c) \), with \( P \) neighbouring pixels \( I^D(x_p, y_p) \) for \( p = 0, ..., P - 1 \) is defined as follows:

\[
LDBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(I^D(x_p, y_p) - I^D(x_c, y_c))
\]  

(3.6)

where

\[
s(v) = \begin{cases} 
1 & \text{if } v \geq 0 \\
0 & \text{otherwise}
\end{cases}
\]

(3.7)

Examples of a 3D facial mesh and depth map and resulting LDBP image for AU28 can be seen in Figs. 3.1a, 3.1b and 3.4a respectively.

3.4.2 Local Azimuthal Binary Patterns

Two new features based on the LBP idea, the Local Azimuthal Binary Pattern - Distance (LABP\(_D\)) and the Local Azimuthal Binary Pattern - Component (LABP\(_C\)), are proposed. These features employ the Azimuthal projection, both in the form of the APDI and APCIs, rather than the depth map, in order to encode the angular normal information in the neighbourhood of each point. In this way, the shape of the mesh is encoded via the direction of the normals at each point, allowing more subtle information to be captured about the structure of the mesh. Let \( I^D \) be the APDI image. Then the LABP\(_D\) operator
for a central point \( I^D(x_c, y_c) \), with \( P \) neighbouring pixels \( I^D(x_p, y_p) \) for \( p = 0, ..., P - 1 \) is defined as follows:

\[
LABP_D(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(I^D(x_p, y_p) - I^D(x_c, y_c)) \tag{3.8}
\]

The \( LABP_C \) operator has two components, and is equivalently formed by applying the binary pattern operator to each of the APCIs, \( I^C_x \) and \( I^C_y \):

\[
LABP_{C_x}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(I^C_x(x_p, y_p) - I^C_x(x_c, y_c)) \tag{3.9}
\]

\[
LABP_{C_y}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(I^C_y(x_p, y_p) - I^C_y(x_c, y_c)) \tag{3.10}
\]

where \( s(v) \) is as defined in the Section 3.4.1.

Examples of the \( LABP_D \) and \( LABP_C \) images resulting from the examples in Figs. 3.1d-3.1f can be seen in Figs. 3.4b-3.4d respectively.

### 3.4.3 Local Normal Binary Patterns

The proposed LNBPs employ the normals of the triangular polygons that form the 3D face mesh to encode the shape of the mesh at each point in an alternative way to LABPs. Again, this employs a richer source of information about the shape of the facial mesh than depth alone. Two feature descriptor types, \( LNBP_{OA} \) and \( LNBP_{TA} \), are proposed, each of which uses the cosine of the angular differences between normals to produce a binary number that encodes the shape of the neighbourhood around the central point, \((x_c, y_c)\).

As with LBPs, a circular neighbourhood is defined around each point, specified by a radius \( r \) and \( P \) points regularly spaced around the circle. The unit normal, \( n_p \), at each point, \((x_p, y_p)\), in the neighbourhood is found, along with that at the central point, \( n_c \), through \( x-y \) interpolation of the given points in the mesh. The \( LNBP_{OA} \) then calculates the scalar product of the two normals, and assigns a one or zero depending on whether this is higher or lower than the scalar product of the central normal, \( n_c \), with a vector that defined to be a given threshold angle, \( \psi \), from the central normal, \( n_c \). This then produces a \( p \)-bit binary number for neighbourhood around each point in the grid. Alternatively,
3.4. Binary Pattern Features

Figure 3.3: LDBP and LNBP_{OA} operation using eight surrounding points. Figs. (a)-(c): The LDBP operation - (a) Original depth values (b) Difference in depth values (c) Resulting binary number. Figs. (d)-(e): LNBP_{OA} operation - (a) Direction of normals (b) Angle difference between normals (c) Resulting binary number.

The LNBP_{TA} calculates the difference in the two angles of the normals, the azimuth and the elevation, and assigns bits depending on how these compare to $\psi_a$ and $\psi_e$ respectively.

The LNBP_{OA} is calculated as follows:

$$LNBP_{OA}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p t(\langle n_c, n_p \rangle, \langle n_c, n_t \rangle)$$  \hspace{1cm} (3.11)

where $\langle n_c, n_t \rangle = \cos(\psi)$ and

$$t(x, \tau) = \begin{cases} 
1 & \text{if } x < \tau \\
0 & \text{otherwise}
\end{cases}$$  \hspace{1cm} (3.12)
3. Binary Pattern Features for 3D Facial Action Unit Detection

The two LNBP\textsubscript{T,A} components are calculated as follows:

\\begin{equation}
\text{LNBP}\textsubscript{T,A}^a(x_c, y_c) = \sum_{p=0}^{P-1} 2^p t(\cos(|\theta_c - \theta_p|), \cos(\psi_a))
\\end{equation}

\\begin{equation}
\text{LNBP}\textsubscript{T,A}^e(x_c, y_c) = \sum_{p=0}^{P-1} 2^p t(\cos(|\phi_c - \phi_p|), \cos(\psi_e))
\\end{equation}

where $\theta = \arctan(\frac{y}{x})$ is the azimuth angle, and $\phi = \arctan(\frac{z}{x})$ is the elevation angle, of a normal $n = xi + yj + zk$. The process of applying the LNBP\textsubscript{OA} operator is shown in Figs 3.3d-3.3f, while an example of the normals used, a zoom in on the normals, and the LNBP\textsubscript{OA} image and the two LNBP\textsubscript{T,A} images can be seen in Figs. 3.2a, 3.2b, 3.4e, 3.4f and 3.4g respectively.

The LNBP\textsubscript{T,A} operator is used to form feature descriptors using two methods:

- Firstly, by forming separate histograms from each angle image and concatenating these to form one descriptor (LNBP\textsubscript{T,A1}).

- Secondly, by forming a 2D histogram from the combination of the two components, and then flattening this histogram to form a 1D descriptor (LNBP\textsubscript{T,A2}). This method is described in more detail in Section 3.5.

3.4.4 Local Phase Quantisers

LPQs can be used to extract the local phase information, which contains important directional features (e.g. edges) [111] useful in analysis of the facial deformations, from the image. They exploit the Short-Term Fourier Transform (STFT) calculated over a rectangular neighbourhood, to find the local frequency information around each point, and encode the resulting phase as a binary number in a similar way to LBPs. They are designed to be particularly invariant to blurring as the phase information they extract is unaffected by an assumed symmetric blur pattern [110].
3.4. Binary Pattern Features

Figure 3.4: Basic and LPQ-based 3D feature images (a) LDBP image (b) LABP$_D$ image (c)-(d) LABP$_C$ $x$ and $y$ images (e) LNBP$_{OA}$ image (f)-(g) LNBP$_{TA}$ azimuthal and elevation images (h) LDPQ image (i) LAPQ image

The general definition of the LPQ is as follows. For a point $\mathbf{p} = (x, y)$ with $M \times M$ neighbourhood $\mathcal{N}_p$, the STFT is defined as:

$$F(\mathbf{u}, \mathbf{p}) = \sum_{\mathbf{k} \in \mathcal{N}_p} f(\mathbf{p} - \mathbf{k})e^{-j2\pi \mathbf{p}^T \mathbf{k}} = \mathbf{w}_u^T \mathbf{f}_p$$

(3.15)

where $\mathbf{w}_u$ is the basis vector of the 2D Discrete Fourier Transform (DFT) at frequency $\mathbf{u}$ and $\mathbf{f}_p$ is the vector form of the neighbourhood $\mathcal{N}_p$. In addition, in this implementation, instead of applying employing uniform windows which truncate the basis functions and would result in a large amount of frequency leakage, a Gaussian window is applied to give smoother cut-off and less leakage. Four frequency pairs $(u_1, ..., u_4)$ are employed, $([0, 0]^T, [0, a]^T, [a, a]^T, [a, -a]^T)$, which results in the following vector at each point:

$$\mathbf{F}_p = [Re\{F(u_1, p)\}, ..., Re\{F(u_4, p)\}, Im\{F(u_1, p)\}, ..., Im\{F(u_4, p)\}]$$

(3.16)

However, it can be shown that quantisation preserves the maximal amount of information when the samples are statistically independent. In both the depth map and APDI representations, as in 2D images, neighbouring pixels will be highly correlated. In order to overcome this, a decorrelation process is applied as in the 2D case. A whitening transform
3. Binary Pattern Features for 3D Facial Action Unit Detection

Figure 3.5: Examples of images produced by the different feature descriptors for subject bs043 performing AU20. (a)-(b) Gabor depth magnitude and phase images (c)-(d) Gabor APDI images (e)-(f) LDGBP magnitude and phase images (g)-(h) LAGBP images (i)-(k) Monogenic depth magnitude, phase and orientation images (l)-(n) Monogenic APDI images (o)-(q) LDMBP magnitude, phase and orientation images (r)-(t) LAMB images
3.4. Binary Pattern Features

is derived from the covariance matrix, \( D \), of \( F_p \), which is calculated as \( WCW^T \), where \( C \) is the correlation matrix of the original image. The whitening transform, \( V^T \), can then be calculated via singular value decomposition of \( D \), and then this can be applied to the components of \( F_p \) to achieve samples that are independent: \( G_p = V^T F_p \).

The LPQ algorithm is applied to the two representations, the depth map and APDI, in order to produce two operators, LDPQs and LAPQs. For the 2D representations this results in two sets of components, \( G_p^D \) and \( G_p^A \), for the depth map and the APDI respectively. A scalar quantiser is then applied to each component of these vectors, and used to create eight-digit binary numbers for the point:

\[
LDPQ(p) = \sum_{j=1}^{8} s(g^D_p(j))2^{(j-1)}
\]

\[
LAPQ(p) = \sum_{j=1}^{8} s(g^A_p(j))2^{(j-1)}
\]

Examples of the resulting LDPQ and LAPQ images can be seen in Figs. 3.4h and 3.4i respectively.

3.4.5 Gabor Binary Patterns

The majority of work using Gabor filters with binary patterns has focussed on the magnitude of the Gabor coefficients (e.g. [214, 197]). However, a number of works have employed the phase successfully for face recognition (e.g. [213, 196], showing that both components contain useful information for facial analysis. For this reason, both components are exploited here by encoding both the magnitude and phase of the coefficients, using both the original LBP algorithm, and a phase variant proposed here.

Two new Gabor-based features, the LDGBP and the LAGBP are introduced. LDGBPs apply the LBP algorithm to Gabor filtered images of the depth map, whereas LAGBPs apply this method to the APDI. Log Gabor filters of various scales and orientations are employed in both cases. The binary pattern algorithm is then applied to each of the resulting magnitude images in order to encode the local structural information further. The resulting magnitude images are \( g^D_M \) and \( g^A_M \) for the depth map and APDI respectively.
Application of the LBP algorithm then gives:

\[
LDGBP_M(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(g^D_M(x_p, y_p) - g^D_M(x_c, y_c))
\] (3.19)

\[
LAGBP_M(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(g^A_M(x_p, y_p) - g^A_M(x_c, y_c))
\] (3.20)

where \(g^D_M(x_c, y_c)\) is the depth map magnitude image at the central point, \(g^D_M(x_p, y_p)\) is the depth map magnitude image at the \(p^{th}\) point in the neighbourhood and \(P\) is the number of points in the neighbourhood. In the APDI case, \(g^A_M(x_c, y_c)\) and \(g^A_M(x_p, y_p)\) are similarly defined.

Due to the circular nature of the phase, a variant of the traditional LBP algorithm is required to encode this information. Here a method is proposed similar to that applied for calculation of the LNBPs. The difference is taken between the phase at the central point and those of the neighbouring points and compared to a threshold phase difference value, \(\psi\), in order to assign a zero or one for each neighbouring point. For these experiments this threshold value was set to be \(\frac{\pi}{4}\). Using this method results in the following:

\[
LDGBP_P(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(\psi - |g^D_P(x_c, y_c) - g^D_P(x_p, y_p)|)
\] (3.21)

\[
LAGBP_P(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(\psi - |g^A_P(x_c, y_c) - g^A_P(x_p, y_p)|)
\] (3.22)

where \(g^D_P(x_c, y_c)\) is the depth map phase image at the central point and \(g^D_P(x_p, y_p)\) is the depth map phase image at the \(p^{th}\) point in the neighbourhood. Again, the APDI equivalents, \(g^A_P(x_c, y_c)\) and \(g^A_P(x_p, y_p)\), are similarly defined.

Examples of LDGBP and LAGBP images produced can be seen in Figs. 3.5e-3.5f, and 3.5g-3.5h respectively.

### 3.4.6 Monogenic Binary Patterns

As outlined in Section 3.3.2, the Monogenic signal is closely related to Gabor filters, but allows features to be extracted in all directions simultaneously at a particular scale. It is employed here in order to propose two features, LDMBPs and LAMBPs, in an analogous
3.5 Methodology

In this section the methodology for comprehensive evaluation and comparison of the different binary pattern features is described. The system used consists of several stages: alignment using facial feature points either provided or manually selected, features extraction, GentleBoost region and feature selection, SVM 5-fold parameter optimisation, training and testing.

Figure 3.6: An overview of the proposed system. This shows the different stages: producing facial geometry representations, filter preprocessing, binary pattern analysis, histogram-based feature descriptor concatenation, and SVM parameter optimisation, training and testing.

way to the Gabor methods. This signal is calculated for both the depth map and APDI, for the LDMPB and LAMBP respectively. The magnitude, phase and orientation images are then encoded using the binary pattern algorithm and phase variation outlined in Section 3.4.5. Examples of LDMPB and LAMBP images, can be seen in Figs. 3.5o-3.5q, and 3.5r-3.5t respectively.
traction using the methods described above, feature selection performed using a two-stage method, and training of binary SVM classifiers. Fig. 3.6 shows an overview of the system. Table 3.1 summarises all proposed binary pattern features and abbreviations for the readers convenience.

Algorithm 1 GentleBoost Feature Selection Algorithm

Require: Set of available features \( F \)

Require: Set of chosen features \( F_c \)

Require: Training examples \( X \) with equivalent labels \( y \)

Require: Max number of features to select \( N \)

Initialise weights \( w \) of all examples to be equal and to sum to 1

while \( n < N \) do

Initialise best error score \( \epsilon_b = \inf \)

for each feature \( f \) in \( F \) do

Calculate \( a_f \) and \( b_f \) such that \( y = w(a_f x_f + b_f) \)

Calculate the regression error for this feature, \( \epsilon_f \)

if \( \epsilon_f < \epsilon_b \) then

Set \( f \) to be best feature \( f_b = f \)

Set \( \epsilon_b = \epsilon_f \)

end if

end for

Calculate estimation of each label

\[ \hat{y} = w(a_{f_b} x_{f_b} + b_{f_b}) \]

Update weights \( w = w e^{-y \hat{y}} \) and renormalise

Remove \( f_b \) from \( F \)

Add \( f_b \) to \( F_c \)

end while

3.5.1 Feature Extraction

The data provided in the two databases differed in format - full meshes are provided for the D3DFACS database, but only point clouds for the Bosphorus dataset. In order

Figure 3.7: Construction of the feature vector by concatenation of histograms from each block in the image, and then concatenation of different image histograms, in this case the magnitude, phase and orientation images for once scale of the LAMBPs.
### Table 3.1: Summary of proposed features and comparison features (shown in italics). This table details the abbreviations, descriptions, feature descriptor lengths, and the number of components, for each feature type employed.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Full Name</th>
<th>Description</th>
<th>Length</th>
<th>#C</th>
</tr>
</thead>
<tbody>
<tr>
<td>2DLBP [109]</td>
<td>2D Local Binary Patterns</td>
<td>Intensity image binary pattern features</td>
<td>6000</td>
<td>1</td>
</tr>
<tr>
<td>2DLGBP [214]</td>
<td>2D Local Gabor Binary Patterns</td>
<td>Binary pattern features from Gabor filtered intensity images</td>
<td>96000</td>
<td>2</td>
</tr>
<tr>
<td>LDBP [61]</td>
<td>Local Depth Binary Patterns</td>
<td>Depth map binary pattern features</td>
<td>6000</td>
<td>1</td>
</tr>
<tr>
<td>LNP [82]</td>
<td>Local Normal Patterns</td>
<td>Binary pattern features from normal components</td>
<td>18000</td>
<td>3</td>
</tr>
<tr>
<td>GMC [143]</td>
<td>Gabor Mean Curvature</td>
<td>Gabor filters applied to mean curvature image</td>
<td>160000</td>
<td>1</td>
</tr>
<tr>
<td>GSI [143]</td>
<td>Gabor Shape Index</td>
<td>Gabor filters applied to shape index image</td>
<td>160000</td>
<td>1</td>
</tr>
<tr>
<td>HNC [70]</td>
<td>Haar Normal Components</td>
<td>The Haar transform applied to the normal components</td>
<td>786432</td>
<td>3</td>
</tr>
<tr>
<td>LABP_D</td>
<td>Local Azimuthal Binary Patterns - Distance</td>
<td>Azimuthal distance binary pattern features</td>
<td>6000</td>
<td>1</td>
</tr>
<tr>
<td>LABP_C</td>
<td>Local Azimuthal Binary Patterns - Components</td>
<td>Azimuthal component binary pattern features</td>
<td>12000</td>
<td>2</td>
</tr>
<tr>
<td>LNBP_OA</td>
<td>Local Normal Binary Patterns (One Angle)</td>
<td>Absolute angle normal binary pattern features</td>
<td>6000</td>
<td>1</td>
</tr>
<tr>
<td>LNBP_TA1</td>
<td>Local Normal Binary Patterns (Two Angle Method 1)</td>
<td>Azimuth and elevation angle normal binary pattern features, 1D histogram</td>
<td>12000</td>
<td>2</td>
</tr>
<tr>
<td>LNBP_TA2</td>
<td>Local Normal Binary Patterns (Two Angle Method 2)</td>
<td>Azimuth and elevation angle normal binary pattern features, 2D histograms</td>
<td>360000</td>
<td>1</td>
</tr>
<tr>
<td>LDPQ</td>
<td>Local Depth Phase Quantiser</td>
<td>Local phase quantisation applied to depth map</td>
<td>6000</td>
<td>1</td>
</tr>
<tr>
<td>LAPQ</td>
<td>Local Azimuthal Phase Quantiser</td>
<td>Local phase quantisation applied to APDI</td>
<td>6000</td>
<td>1</td>
</tr>
<tr>
<td>LDGBP</td>
<td>Local Depth Gabor Binary Patterns</td>
<td>Binary pattern features from the Gabor filtered depth map</td>
<td>192000</td>
<td>2</td>
</tr>
<tr>
<td>LAGBP</td>
<td>Local Azimuthal Gabor Binary Patterns</td>
<td>Binary pattern features from the Gabor filtered APDI</td>
<td>192000</td>
<td>2</td>
</tr>
<tr>
<td>LDMBP</td>
<td>Local Depth Monogenic Binary Patterns</td>
<td>Binary pattern features from the Monogenic filtered depth map</td>
<td>72000</td>
<td>3</td>
</tr>
<tr>
<td>LAMBP</td>
<td>Local Azimuthal Monogenic Binary Patterns</td>
<td>Binary pattern features from the Monogenic filtered APDI</td>
<td>72000</td>
<td>3</td>
</tr>
</tbody>
</table>
3. Binary Pattern Features for 3D Facial Action Unit Detection

**Algorithm 2** GentleBoost Region Selection Algorithm

**Require:** Set of available regions $R$ of features

**Require:** Set of chosen regions $R_c$ of features

**Require:** Training examples $X$ with equivalent labels $y$

**Require:** Max number of regions to select $N$

Initialise weights $w$ of all examples to be equal and to sum to 1

while $n < N$ do

Initialise best error score $\epsilon_b = \infty$

for each region $r$ in $R$ do

Set the total error $\epsilon_r = 0$

for each feature $f$ in region $r$ do

Calculate $a$ and $b$ such that

$$y = w(a_fX_f + b_f)$$

Calculate the regression error for this feature, $\epsilon_f$

Add to the total error $\epsilon_r = \epsilon_r + \epsilon_f$

end for

if $\epsilon_r < \epsilon_b$ then

Set $r$ to be best region $r_b = r$

Set $\epsilon_b = \epsilon_r$

end if

end for

Calculate averaged estimation of each label

$$\hat{y} = \frac{1}{F}\sum_{f \in r_b} w(a_fX_f + b_f)$$

Update weights $w = we^{-y\hat{y}}$ and renormalise

Remove $r_b$ from $R$

Add $r_b$ to $R_c$

end while

to ensure that both databases are in the same form for comparison, the first processing stage involves surface triangulation of the points to create a mesh in the case of Bosphorus, and appropriate Laplacian smoothing for both databases. As the data in the Bosphorus database is more noisy than that of the D3DFACS, more smoothing is required in this case. The Visualisation Toolkit (VTK) library was employed for the triangulation, smoothing, and to extract the normals directly from the resulting meshes.

In order to extract corresponding features across the datasets, first the 3D facial meshes must be aligned in the $x$-$y$ plane, via the eye, nose and chin facial feature points. These were provided with the Bosphorus database [139] and manually selected in the D3DFACS database [35]. They are used to apply a similarity transform to the faces, consisting of translation, scaling, and in the case of the D3DFACS database, rotation, to align the
3.5. Methodology

eyes and nose of all the facial meshes. The distance between the eyes was set to 125 pixels. Rotation is not required for the Bosphorus database as the faces are already uniformly oriented. Once alignment is complete, the four facial geometry representations are calculated. The depth map is determined by interpolation of the $z$ values across a regular $x$-$y$ grid, while a similar interpolation is performed on the three components of the normals extracted from the facial mesh to form three regular grids. These are used directly for LNBP calculation, and further processed to calculate the APDI for the facial geometries.

Features are then extracted using all of the methods outlined in Section 3.4. The images are divided into equally sized square regions, and histograms are formed from the binary numbers calculated within each of these regions, using a number of bins to divide up the full range of numbers possible. For all features except LNBP$_{TA^2s}$, these small histograms are concatenated into one large feature vector, to produce the descriptor for that feature type. This process can be seen in Fig. 3.7. In the case of the Gabor and Monogenic features, because there are respectively 2 or 3 separate components to consider, these can either be combined in two ways, as will be described in the Section 3.5.4. For LNBP$_{TA^2s}$, the approach is different, as the two components are highly related. Here, the binary numbers are used to construct a 2D histogram, using 60 bins in each direction, and then the result is flattened to create the 1D feature descriptor. This process results in a much longer descriptor than that of the LNBP$_{OA}$ or LNBP$_{TA^1}$. The size of the feature descriptor, including the length of each component, and number of components, can be seen in Table 3.1. This table demonstrates the large differences in feature descriptor lengths, and shows why a compromise may sometimes be desired between accuracy and computation time.

3.5.2 Feature Selection

Feature selection is performed using a two-stage GB algorithm. First, regions are selected which contain the most discriminative information for the AU within the feature descriptor. Then individual features are selected from within these chosen regions. This region selection step allows features in parts of the face where the AU is not active to
3. Binary Pattern Features for 3D Facial Action Unit Detection

Figure 3.8: The full feature selection process. (a) The feature descriptor formed through concatenation of small histograms from each region (b) Regions are chosen based on the combined benefit of the features in the small histograms (c) Individual features are then chosen from within the chosen regions.

be discarded quickly before the main feature selection step, thus greatly reducing the computation time taken for feature selection.

**Individual GentleBoost**  The feature selection algorithm employed is the GB algorithm. Each feature is taken in turn used to calculate a least-squares regression on the training data and labels. The feature that produces the lowest error rate is chosen as the most discriminative in this pass. Then the predicted labels when exploiting this feature is used to update the weights for all examples in the training set, allowing examples that
are misclassified to be focussed on (boosted) in the next pass as they become more highly weighted. This can be seen in full in Algorithm 1.

**Region Selection GentleBoost** The region selection algorithm is based on the individual feature selection GB algorithm, and can be seen in Algorithm 2. Each region is taken in turn and the individual features extracted from this region are used to calculate a least-squares regression on the training data and labels. The error is summed for all regions, and the one that produces the lowest error rate is chosen as the most discriminative in this pass. Then the predicted labels when using the features in this region are averaged to give an estimated label, and this is used to update the weights for all examples in the training set, allowing examples that are misclassified to be focussed on (boosted) in the next pass as they become more highly weighted.

The second stage is application of the feature-wise GB selection to choose particular features within these regions. To avoid over-fitting, the strategy here is to run this stage of the selection algorithm repeatedly, removing the previously chosen features at each stage, until the number of features selected exceeds the number of examples in the training set, or until fewer than 5 features are being chosen by the algorithm.

**Performance** In order to assess the effectiveness of this approach, experiments were conducted to compare the performance, and time taken, for the feature selection both with, and without, region selection. For these experiments, the \( \text{LABP}_D \) operator was employed, and experiments were conducted on a subset of AUs. The performance results of this can be seen in Table 3.2, along with the comparative running times for each feature selection method for a typical fold, as well as the number of features selected by each method. The computational times are calculated on a quad-core 3.2GHz computer, running the algorithm in MATLAB. These results show how for both operators the region selection algorithm achieves a comparable, or higher, performance for all AUs, and on average on this subset. It also greatly reduces the computational expense of the feature selection procedure for almost every case shown here, only taking more time for one AU, and in addition a comparable number of features are chosen in almost all cases. Therefore, these results demonstrate that region selection can greatly speed up the process of feature
3. Binary Pattern Features for 3D Facial Action Unit Detection

<table>
<thead>
<tr>
<th>AU</th>
<th>Individual Selection</th>
<th>With Region Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROC (%)</td>
<td>Time (s)</td>
</tr>
<tr>
<td>1</td>
<td>95.41</td>
<td>2064</td>
</tr>
<tr>
<td>7</td>
<td>94.97</td>
<td>3594</td>
</tr>
<tr>
<td>10</td>
<td>94.37</td>
<td>788</td>
</tr>
<tr>
<td>14</td>
<td>91.53</td>
<td>500</td>
</tr>
<tr>
<td>15</td>
<td>94.18</td>
<td>718</td>
</tr>
<tr>
<td>16</td>
<td>83.84</td>
<td>759</td>
</tr>
<tr>
<td>23</td>
<td>82.09</td>
<td>743</td>
</tr>
<tr>
<td>μ</td>
<td>90.91</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Validation of the region selection method

selection, whilst maintaining performance.

3.5.3 Classification

Binary SVM classifiers are trained to detect each of the AUs. SVMs is a supervised classification method that finds a decision boundary that maximises the perpendicular distance between itself and a set of chosen support vectors which are the closest examples from each class.

Let \( \{x_1, ..., x_M\} \) be the input training data for \( M \) frames, with corresponding labels, \( \{y_1, ..., y_M\} \), for each AU, which take the form of \( y_i \in \{-1, +1\} \). The simple form of the SVM problem is then described by the following constraints:

\[
\begin{align*}
w^T x + b & \geq +1 \quad for \quad y_i = +1 \\
w^T x + b & \leq -1 \quad for \quad y_i = -1 
\end{align*}
\]

(3.23)

where \( w \) is a coefficient weight vector for the hyperplane dividing the two classes, and \( b \) is a threshold. This can be combined into one set of inequalities as follows:

\[
y_i (w^T x_i + b) - 1 \geq 0 \quad for \ all \ i
\]

(3.24)

The optimal weights are found through solving the following optimisation problem:

\[
\begin{align*}
\text{minimise} & \quad \frac{1}{2} ||w||^2 + C \sum_{i=1}^{M} \xi_i \\
\text{subject to} & \quad y_i (w^T x_i - b) \geq 1 - \xi_i, \\
& \quad \xi_i \geq 0
\end{align*}
\]

(3.25)

where \( \xi_i \) is the slack variable for data \( x_i \) that violates this condition. In practise, the feature vector can be transformed into a high dimensional space through the use of non-
3.5. Methodology

linear functions. And in the the dual formulation this can be done through employing kernel functions to measure similarity between the training examples. Here, due to the histogram form of features, the histogram intersection function is employed for this purpose, which takes the form of \( k(x_i, x_j) = \sum_{n=1}^{N} \min(x_{in}, x_{jn}) \). Parameter optimisation is performed via 3-fold cross-validation on a validation set, and the SVM classifiers are then trained on a separate training set.

3.5.4 Feature Fusion

A number of feature types were found to show complementary behaviour in the detection of AUs. In order to benefit from the most useful aspects of both features, and improve overall performance, fusion of one of more feature type is explored. Two alternative methods were trialled during the component testing of the Gabor and Monogenic feature types, which will be described in Section 3.6.1, feature level and classification level fusion.

**Feature level fusion** Here the descriptors for one or more feature are concatenated before feature selection, allowing the algorithm to choose the best features from all of those available.

**Classification level fusion** This method takes the positive and negative testing confidence levels of each of the individual classifiers and averages these, in order to calculate new overall confidence levels (for both the positive and negative classes) that can then be used to determine the classification of each example in the testing set. This method can give a different result to majority voting if the confidence levels are close to 0.5. For example, see the results from testing three classifiers on two examples, shown in Table 3.3. In this case it can be seen how majority voting would give a positive classification in both cases. However, using the proposed method this will give respective averages of 0.51 and 0.44, which will results in positive classification in the first case, and negative in the second. This method of classification level fusion was shown to be more reliable than feature level, and was thus used at later stages for combinations of other feature types.
3. Binary Pattern Features for 3D Facial Action Unit Detection

<table>
<thead>
<tr>
<th>Feature</th>
<th>Confidence</th>
<th>Class</th>
<th>Confidence</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pos</td>
<td>Neg</td>
<td>Pos</td>
<td>Neg</td>
</tr>
<tr>
<td>2DLBP</td>
<td>0.34</td>
<td>0.66</td>
<td>Neg</td>
<td>0.56</td>
</tr>
<tr>
<td>3DLBP</td>
<td>0.55</td>
<td>0.45</td>
<td>Pos</td>
<td>0.52</td>
</tr>
<tr>
<td>LABP_D</td>
<td>0.65</td>
<td>0.35</td>
<td>Pos</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.76</td>
</tr>
<tr>
<td>Majority vote</td>
<td>-</td>
<td>-</td>
<td>Pos</td>
<td>-</td>
</tr>
<tr>
<td>Class fusion</td>
<td>0.51</td>
<td>0.49</td>
<td>Pos</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.56</td>
</tr>
</tbody>
</table>

Table 3.3: Example of the results given by majority voting and the proposed classification level fusion method for two different samples.

3.6 Experiments

A variety of experiments were conducted to explore the performance of the new feature types. These aimed to identify the optimal parameter settings, determine the benefit of the proposed 3D features over previous techniques (such as 2D Local Binary Patterns (2DLBPs) and LDBPs) and to demonstrate how robust they are to variations in data through cross-database testing. For this purpose, the only two 3D facial databases that include FACS coding were used: the Bosphorus database [139], and the D3DFACS database [35]. The former consists of static images displaying 105 subjects performing combinations of AUs and the six basic expressions, as well as neutral examples, fully FACS coded with intensities. The D3DFACS database consists of videos of 10 subjects recorded performing combinations of AUs. Each sequence has been assigned a single FACS code that corresponds to the AUs present in the peak frame of video, but no intensities are included.

Experiments conducted on the Bosphorus database were performed using 10-fold cross validation. The 10 folds were first created by dividing the randomly ordered subjects into equal sized groups. Nine of these were then used to form balanced training sets composed of all positive examples that demonstrated the AU at an intensity of C or above from each subject, plus an equal number of negative examples of any intensity. Intensity examples for a particular AU of A or B intensity were removed in the training set for that AU, though left in as negative examples for other AUs.

The full dataset division strategy is as follows. After dividing the subjects into 10 folds, one fold is taken as the test set, and all examples are taken from every subject and set aside for testing. The remaining folds are employed to create training and validation sets.
for each AU.

Firstly, all available positive examples are taken from each subject, and then an equal number of negative examples are chosen at random from the remaining examples for each subject. A validation set is then taken from these examples. As the validation set must be neither too large (as this would make parameter optimisation very slow) nor so small that optimisation is not possible, the percentage of the training set that is used for parameter optimisation is allowed to vary according to the total size of the training set. A larger percentage is taken when there are only a small number of examples in the training set, and a smaller percentage when this number is large. Therefore the percentage of subjects taken from all those in the 9 training folds, \( p \), of the number of training examples, \( N \), was chosen to be as follows:

\[
p = \begin{cases} 
0.1 & N > 500 \\
0.2 & 100 < N < 500 \\
0.25 & N < 100 
\end{cases} \tag{3.26}
\]

The examples from the remaining subjects are taken as the training set.

In all of the experiments conducted in this section the ROC AuC was calculated for each fold individually, and these results were then averaged using a weighted mean (according to number of examples) to find a final score for each AU, and mean operator performance.

### 3.6.1 Parameter testing

Experiments were conducted to explore the optimal parameter settings for each of the feature types. For this purpose experiments were run over a subset of AUs: 1, 7, 10, 14, 15, 16, 23. These were chosen as they cover the full face region, and cover a range of shapes that can be difficult to detect accurately, for example: narrowing of the eyes and straightening of the lower eyelid that distinguishes 7 from 6, cheek dimples that distinguish 14 from 12, 13 and 20, and tightening of the lips that distinguish 23 from 24 and the number of AUs that stretch the lips. Thus they can be taken as representative subset of AUs that are likely to display differences in performance as the parameters are varied. The tests conducted included: different radii values, threshold values for LNBPš, different numbers of image regions and histogram bins, and individual component perfor-
### 3. Binary Pattern Features for 3D Facial Action Unit Detection

<table>
<thead>
<tr>
<th>Operator</th>
<th>Mean Radius Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDBP</td>
<td>88.24 89.28 89.93 90.26 89.86</td>
</tr>
<tr>
<td>LABP</td>
<td>90.94 90.97 91.45 91.89 91.68</td>
</tr>
<tr>
<td>LDGBP</td>
<td>93.39 93.79 93.78 94.19 93.80</td>
</tr>
<tr>
<td>LAGBP</td>
<td>92.14 92.09 92.60 92.43 93.12</td>
</tr>
<tr>
<td>LDMBP</td>
<td>91.17 93.05 93.73 93.98 94.34</td>
</tr>
<tr>
<td>LAMBP</td>
<td>91.49 91.39 92.37 92.54 92.46</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threshold</th>
<th>LNBP$_{OA}$</th>
<th>Mean Radius Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>π/4</td>
<td>82.25 87.00 89.93 93.38 93.16</td>
<td></td>
</tr>
<tr>
<td>π/8</td>
<td>83.62 89.64 92.20 92.69 92.58</td>
<td></td>
</tr>
<tr>
<td>π/16</td>
<td>83.73 90.08 92.07 92.72 91.96</td>
<td></td>
</tr>
<tr>
<td>π/32</td>
<td>86.30 89.84 91.44 91.95 91.43</td>
<td></td>
</tr>
</tbody>
</table>

### Neighbourhood Size Results

<table>
<thead>
<tr>
<th>Operator</th>
<th>3 7 11 15 21</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDPQ</td>
<td>92.44 93.92 94.03 93.76 93.64</td>
</tr>
<tr>
<td>LAPQ</td>
<td>92.41 91.91 92.01 92.23 92.22</td>
</tr>
</tbody>
</table>

Table 3.4: Parameter testing results for each of the new features.

<table>
<thead>
<tr>
<th>AU</th>
<th>Number of Regions</th>
<th>Number of Bins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 10 12 15 30 60 90 120</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>95.8 95.7 96.2 96.5 95.4 95.7 96.0 95.7</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>94.2 95.0 95.2 95.7 95.2 95.0 95.2 95.7</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>95.4 94.3 94.7 94.5 94.7 94.3 94.6 95.1</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>93.9 95.5 95.1 93.7 92.5 95.5 93.2 93.4</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>92.9 94.1 93.3 93.3 92.6 94.1 93.8 93.8</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>85.1 84.5 82.3 83.8 82.2 84.5 84.9 84.9</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>84.4 84.2 85.1 84.7 83.0 84.2 83.5 83.4</td>
<td></td>
</tr>
<tr>
<td>μ</td>
<td>91.7 91.9 91.7 91.8 90.8 91.9 91.6 91.7</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.5: Results of varying the number of regions and bins for feature extraction.

In the parameter tests, for simplicity, the (ROC AuC was calculated from the full set of results across all ten folds.

**Radius Testing** The mean results of the radius tests can be seen in Table 3.4. Tests were conducted across the subset of AUs for 5 difference radii values: 1, 3, 5, 8 and 11. These show that a radius of 8 is the optimal for both the LDBP and LABP feature types, as well as the LDGBP and LAMBP operators. However, LAGBPs and LDMBPs produce a slightly higher result with a radius of 11. Not shown in this table are the individual AU results which show a wide variation in performance. However, for the majority of these operators, the radius that results in the highest overall performance, also produces a comparable result for most AUs to the maximum achieved score. LDPQs
### 3.6. Experiments

<table>
<thead>
<tr>
<th>AU</th>
<th>LDGBP</th>
<th>LDMBP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mag Phase</td>
<td>Feat Class</td>
</tr>
<tr>
<td>1</td>
<td>97.5</td>
<td>97.6</td>
</tr>
<tr>
<td>7</td>
<td>96.6</td>
<td>97.4</td>
</tr>
<tr>
<td>10</td>
<td>96.4</td>
<td>96.1</td>
</tr>
<tr>
<td>14</td>
<td>96.8</td>
<td>96.3</td>
</tr>
<tr>
<td>15</td>
<td>93.0</td>
<td>93.4</td>
</tr>
<tr>
<td>16</td>
<td>90.8</td>
<td>89.7</td>
</tr>
<tr>
<td>23</td>
<td>88.6</td>
<td>82.8</td>
</tr>
<tr>
<td>μ</td>
<td>94.2</td>
<td>93.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AU</th>
<th>LAGBP</th>
<th>LAMBP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mag Phase</td>
<td>Feat Class</td>
</tr>
<tr>
<td>1</td>
<td>96.6</td>
<td>96.7</td>
</tr>
<tr>
<td>7</td>
<td>96.8</td>
<td>97.1</td>
</tr>
<tr>
<td>10</td>
<td>97.0</td>
<td>96.2</td>
</tr>
<tr>
<td>14</td>
<td>93.8</td>
<td>95.4</td>
</tr>
<tr>
<td>15</td>
<td>92.7</td>
<td>93.6</td>
</tr>
<tr>
<td>16</td>
<td>85.7</td>
<td>88.4</td>
</tr>
<tr>
<td>23</td>
<td>83.2</td>
<td>83.9</td>
</tr>
<tr>
<td>μ</td>
<td>92.3</td>
<td>93.0</td>
</tr>
</tbody>
</table>

Table 3.6: Component and fusion test results for LDGBPs, LAGBPs, LDMBPs and LAMBPs.

and LAPQs were tested using a range of different neighbourhood sizes. The range used here was: 3, 7, 11, 15 and 21. The results show that apart from a much lower result for a neighbourhood of 3 for LAPQs, all neighbourhood sizes achieved similar results for these operators, showing no significant performance differences. Hence, the neighbourhoods that showed highest average results, and which also demonstrated good performance on all the individual AUs, were taken as the optimal values. These were neighbourhoods of 11 and 3 respectively for the LDPQ and LAPQ features.

**Threshold angle for LNBP**s  The LNBP operators have two parameters that can be varied, the radius and the threshold angle, which affect how the feature behaves in different ways, and can counteract the effects of each other. For this reason, tests were conducted to explore the performance variation across different values for both of these parameters. The radii tested were the same as those used for other features (except LPQs), and the threshold angles tested were \( \left( \frac{\pi}{12}, \frac{\pi}{18}, \frac{\pi}{24}, \frac{\pi}{30} \right) \). As can be seen from the results, for all threshold values the highest average results were achieved with a radius of 8, and the best overall score was with this radius and a threshold of \( \frac{\pi}{12} \). However, interestingly, it is with this threshold that the worst average score is also achieved, with a radius of 1. This can
be explained by the fact that with a bigger threshold angle it would be expected that the number of points that have a normal outside of the threshold will change dramatically as the radius changes, as almost all points will be inside that threshold when the radius is 1, but far fewer with a larger radius. However, when the threshold angle is very small, regardless of the radius most points will have a normal outside of it, and so a huge change is not expected in the results achieved with different radii. This is demonstrated in the \( \frac{\pi}{30} \) results, which do not vary greatly across the different radii values.

**Number of regions and bins** In order to assess the optimal number of regions and bins for extracting the feature descriptors experiments were conducted, using the \( \text{LABP}_D \) feature with a radius of 8, where each parameter was varied, while the other kept constant. The default number of regions (while varying bin number) was 10, and the default number of bins 60. The results of this can be seen in Table 3.5. These results show how the number of regions does not greatly affect the average performance: The optimal number varies for different AUs, but generally a similar result is achieved, regardless of the number of regions. As having 10 regions achieves a marginally superior average performance, and the result is not significantly decreased from the optimal for any of the AUs, this parameter was taken for all further experiments. For the number of bins, more variation is seen, with 30 bins in particular showing a inferior performance to the other potential values. 60 bins gives the highest average performance, and also outperforms the other numbers on the most AUs. Therefore, this parameter was set thus for the remaining experiments.

**Gabor and Monogenic components** To investigate which aspects of the Gabor and Monogenic binary pattern features are most useful, separate experiments were conducted using each component alone to extract features. For LDGBPs/LAGBPs this meant extracting only magnitude and phase features, and for LDMBP/LAMBPs magnitude, phase and orientation. In addition, the results from each of these components was then fused using the two methods outlined in Section 3.5.4. As feature selection creates the biggest bottle neck in the training process, and the complexity of this is \( \mathcal{O}(n^2) \), where \( n \) is the length of the feature vector, the second method greatly reduces the amount of time taken for training. The results of these tests can be seen in Table 3.6. Firstly these results demonstrate how classifier fusion generally produces either comparable, or superior
results, to feature fusion, and yields a higher average result across this subset of AUs on all four feature types. This is a desirable result, as this method has the benefit that it is much faster to compute due to the much shorter feature descriptors. Secondly, these results show how the discriminative information from the images is distributed between all two or three components in each of the features, although there is a general trend in each feature that one component generally achieves the highest result for most AUs - the magnitude component for LDGBPs, phase for LAGBPs, orientation for LDMBPs and phase for LAMBPs. However, the results show that the maximum performance is always achieved through combining the components. This shows that important information is captured in all components, and thus classifier level fusion is the best method to employ.

3.6.2 Full AU Performance

The mean results for each set of basic and filter-based feature type can be seen in Fig. 3.9, with selected individual AU results for the best performing single features shown in Fig. 3.10. The basic feature types are compared to the previously proposed binary pattern features 2DLBPs [109], LDBPs [61] and Local Normal Patterns (LNPs), a single resolution form of the MS-LNP [82]. The filter-based features are compared to the 2D Local Gabor Binary Pattern (2DLGBP) feature. Also included in Figs. 3.9 and 3.10 are the results from fusion of the LABP_D and LNBP_OA feature types, to show how these compare to the LNP. This is significant as even when combined, these two descriptors are only two thirds of the length of the LNP descriptor.

Figs. 3.9a-3.9b demonstrate how, amongst the basic features, LABP_C and the LABP_D+LNBP_OA combination give the best results on average, outperforming the 2DLBP and LDBP, and achieving comparable results to the LNP. As all of these features, save LNBP_{TA2}, are significantly shorter in length than LNPs, this demonstrates how a more compact representation of the normal information is still able to capture a large amount of the discriminant behaviour necessary for AU detection. In the filter-based features it is the LDGBP, LAGBP and LDMP features that perform best, though LDPQ also do fairly well. The fact that it is possible to achieve a similar result to the Gabor-based features with the Monogenic filters, and to a lesser extent with the phase
3. Binary Pattern Features for 3D Facial Action Unit Detection

Figure 3.9: Mean results for all proposed features, with previous comparisons. (a) Basic feature descriptors. (b) Filter-based feature descriptors.
3.6. Experiments

Figure 3.10: Results for selected features for a number of upper and lower face AUs. AUs 9 and 10 are included as upper face for this comparison as they share characteristics with the other upper face AUs, though they are official labelled as lower face. (a) Upper face basic features. (b) Upper face filter-based features. (c) Lower face basic features. (d) Lower face filter-based features.

quantisers, is an important result as these approaches lead to a much shorted feature descriptor, which saves hugely on processing time and storage whilst achieving a similar performance.

However, more interesting than the mean results are the individual AU results. These demonstrate that, whilst on upper face AUs (Figs. 3.10a-3.10b) 2D features do comparable well or better than 3D, on lower face AUs (Figs. 3.10c-3.10d) the 3D features in general tend to outperform the 2D features, in both the basic and filter-based cases. This result could be due to the differences that are generally observable in the appearance of upper and lower face AUs. In the former case the appearance is often characterised by lines appearing in the face, as can be seen in Fig.3.11. However, in the latter case it is more subtle shapes, particularly around the mouth, that are crucial in distinguishing between
3. Binary Pattern Features for 3D Facial Action Unit Detection

<table>
<thead>
<tr>
<th>Feature</th>
<th>3D Features</th>
<th>Combined with 2D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C-E</td>
<td>B</td>
</tr>
<tr>
<td>2DLBP</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2DLGBP</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LBP</td>
<td>93.3 ±1.0</td>
<td>85.4 ±1.5</td>
</tr>
<tr>
<td>LBP_D</td>
<td>93.9 ±1.0</td>
<td>84.7 ±1.4</td>
</tr>
<tr>
<td>LBP_C</td>
<td>95.3 ±0.7</td>
<td>86.8 ±1.0</td>
</tr>
<tr>
<td>LNBP_OA</td>
<td>92.9 ±0.9</td>
<td>84.2 ±1.8</td>
</tr>
<tr>
<td>LNPB_TA1</td>
<td>93.8 ±0.4</td>
<td>84.9 ±1.0</td>
</tr>
<tr>
<td>LNPB_TA2</td>
<td>93.8 ±0.5</td>
<td>84.6 ±1.0</td>
</tr>
<tr>
<td>LBP_D+LNBP_OA</td>
<td>95.1 ±0.7</td>
<td>86.4 ±1.6</td>
</tr>
<tr>
<td>LDPQ</td>
<td>95.1 ±1.0</td>
<td>86.9 ±1.0</td>
</tr>
<tr>
<td>LAPQ</td>
<td>95.5 ±0.6</td>
<td>87.5 ±1.2</td>
</tr>
<tr>
<td>LDGBP</td>
<td>94.8 ±0.6</td>
<td>87.1 ±1.5</td>
</tr>
<tr>
<td>LGBP+LNBP_OA</td>
<td>94.9 ±0.6</td>
<td>86.9 ±0.9</td>
</tr>
<tr>
<td>LGBP+LNBP_TA2</td>
<td>94.8 ±0.7</td>
<td>86.8 ±1.1</td>
</tr>
<tr>
<td>LDPQ</td>
<td>95.4 ±0.7</td>
<td>86.9 ±1.0</td>
</tr>
<tr>
<td>LDGBP</td>
<td>93.8 ±0.7</td>
<td>84.4 ±1.7</td>
</tr>
<tr>
<td>LGBP</td>
<td>96.0 ±0.8</td>
<td>89.0 ±1.1</td>
</tr>
<tr>
<td>LGBP</td>
<td>95.4 ±0.9</td>
<td>87.7 ±1.3</td>
</tr>
<tr>
<td>LDBP</td>
<td>96.3 ±0.6</td>
<td>88.7 ±1.3</td>
</tr>
<tr>
<td>LAMBP</td>
<td>94.9 ±0.6</td>
<td>86.8 ±1.2</td>
</tr>
</tbody>
</table>

Table 3.7: Mean cross-validation results ($\mu \pm \sigma$) for all experiments. All experiments trained on C-E, tested on intensities shown: Column 1 - 3D features. Column 2 - results from fusion of 2D and 3D features (basic + 2DLBPs and filtered + 2DLGBP). Different AUs. Therefore it is unsurprising that the information provided by 3D data is better at detecting these AUs. Full results for all AUs, along with a more extensive discussion of particular AUs, can be seen in the Appendix.

The results shown here demonstrate that both sets of features appear to be complementary in the AUs they can detect accurately, and it is this behaviour that it is hoped will be harnessed in Section 3.6.3.

Analysis and Comparison In addition to the comparisons carried out in the previous sections, a reduced set of experiments were also conducted on other previously proposed feature types not based on binary patterns, for comparison with the proposed features.

These were Gabor filters applied to the mean curvature and shape index of the depth map, referred here respectively as Gabor Mean Curvature (GMC) and Gabor Shape Index (GSI), proposed by Savran et al. [143], and the Haar transform applied to the normal components, referred to here as Haar Normal Components (HNC), proposed by Kakadiaris
3.6. Experiments

Figure 3.11: Facial action unit examples from the Bosphorus database. (a)-(f)(m)-(r) 2D intensity images (g)-(l)(s)-(x) equivalent 3D facial geometries (a)(g) AU2 (b)(h) AU6+AU7 (c)(i) AU5 (d)(j) AU43 (e)(k) AU14 (f)(l) AU15 (m)(s) AU16 (n)(t) AU20 (o)(u) AU22 (p)(v) AU23 (q)(w) AU24 (r)(x) AU 28.
3. Binary Pattern Features for 3D Facial Action Unit Detection

In the case of the former, in order to conduct a fair comparison, a reduced set of scales and orientations were employed for the Gabor filters, which are the same as those employed for the binary pattern features. Fusion of the two features (GMC and GSI) was performed at the classifier level, as this combination was found to give the highest performance in [143]. The experiments conducted consisted of testing the previously employed subset of representative AUs in the Bosphorus database, as well as conducting cross-database tests using these features. The results of the cross-validation results on the AU subset can be seen in Fig. 3.12, alongside the best performing proposed features. Here it can be seen that, of these AUs, the binary pattern filter-based features outperform all of these features in the majority of cases, and on average. This suggests that, using either Haar transforms, or mean curvature of shape index with this reduced set of Gabor filters, is not able to complete with binary pattern methods of encoding either the depth or azimuthal information.

3.6.3 Fusion Results

A variety of feature combinations were explored through fusion tests, where the output of two or more classifier was combined, to assess the complementary nature of the different feature types, particularly 2D plus 3D. In order to assess the significance at the 5% level of the most promising results over the 2D features alone, t-tests were conducted,
3.6. Experiments

with the results of these included in the discussion below. Experiments were conducted which combined 2D and 3D LBPs with LABP and LNBP feature types. The results of these experiments can be see in Table 3.7. The results show that the features are highly complementary, with significant increases in performance when combining LDBPs with all of the other basic features, due to the fact they do well on different sets of AUs. In addition, combining 2D and 3D features increases average performance above that of any individual operator. For example, when combining LDBPs and LABPs, for all AUs the resulting rates are at least comparable to the highest of the two scores achieved with single feature types. Furthermore, there are a number of AUs (5, 6, 17, 18, 22, 24) for which the resulting rate is higher than either feature type achieved alone. This shows that the combined confidence values from the two features are better able to predict the presence of an AU than when used alone.

Similarly, combining the depth and azimuthal LPQ, Gabor and Monogenic features, was explored. However, this time only minor improvements in the average rate were seen, with the LPQ features showing the most complementary behaviour. The combination of 2D LGBP features with each of these feature types was then examined, the results of which can be seen in Table 3.7. This gives a large increase in average rates, and shows that these features are highly complementary as in the basic feature case. The highest overall average result was 97.0 AuC, and was achieved when 2D LGBPs were combined with both LDGBPs and LDMBPs, though a comparable result was also achieved with 2D LGBPs combined with both LDPQs and LDGBPs (96.9). All of these results demonstrate a significant improvement over 2D LGBPs alone. The azimuthal feature types gave slightly lower results when combined with 2D LGBPs: LAGBPs achieved a significant improvement over 2D LGBPs alone, with a mean score of 96.6. LAPQs and LAMBPs only showed a slight improvement over the 2D features, with scores of 96.4 and only 96.2 respectively. The maximum results achieved here for both 3D features alone (96.3), and 2D/3D fusion (97.0), are comparable to the full results reported in [143], 96.3 and 96.9 respectively for GMC+SI features alone and combined with 2D features, the previous state-of-the-art in AU detection on the Bosphorus database, and where they also conducted experiments using only the intensity values C-E. This is an important result, particularly as the proposed feature descriptors are vastly shorter in length than
3. Binary Pattern Features for 3D Facial Action Unit Detection

those employed when using their features.

3.6.4 Low Intensity AU Testing

As the Bosphorus database provides the full FACS coding for the majority of images, including intensity value, this was employed to run a variety of intensity range combination cross-validation tests. The results from these can be seen in the first two columns of Table 3.7. The format of this test was taken from [143], although in that work only the B intensity examples were employed for testing. The training data used remained the same for all tests, and consisted of all positive examples of intensity C or higher, and negative examples of any intensity. However, here the testing set was composed of positive examples with lower intensity: firstly level B only, and then level A only. The latter test is extremely challenging, as the A level can only be coded by very experienced human coders as it is so difficult to detect. The results of these experiments were compared to the original testing conducted with range C-E. In addition, the combination of 2D features and 3D features were also tested for these intensities.

As would be expected, the results on the lower intensity AUs are reduced from those on the original range. However, a consistent pattern can be observed across the feature types. At both intensity levels, as well as when combined with 2D features, the LDGBPs and LDMBPs show the highest performance on average across the AUs. LDPQs, LAGBPs, and LAMBPs also show some robustness, and demonstrate an improvement at both intensity levels over 2D alone when the two are combined. However, LAPQs do not show the improvement, giving only a very slight increase at the lower intensities when combined with 2DLGBPs. The LDGBP and LDMBP results seen for intensity B show a significant increase over those achieved in [143]. There they reported a maximum ROC AuC of 83.6 with 3D features alone, and 85.4 with 2D and 3D fusion, compared to the maximum results presented here of 89.2 and 90.6 respectively. For the basic features, though all of the new features only produce comparable results to the 2DLBPs when tested alone on the B and A intensity examples, the increases seen in the combination results show that they are all still contributing useful information that is not captured by the 2D features alone. In addition, they again achieve a comparable score to the maximum achieved with
3.6. Experiments

### Table 3.8: Mean cross-database results. Experiments trained on intensities shown: Column 1 - testing on high intensity frames. Column 2 - testing on all frames.

<table>
<thead>
<tr>
<th>Feature</th>
<th>High Intensity</th>
<th>All Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C-E</td>
<td>B-E</td>
</tr>
<tr>
<td>LDBP</td>
<td>86.4 85.6 85.9</td>
<td>84.0 83.2 83.7</td>
</tr>
<tr>
<td>LNP</td>
<td>87.4 86.5 85.2</td>
<td>85.2 83.7 82.4</td>
</tr>
<tr>
<td>GMC</td>
<td>82.2 81.6 79.7</td>
<td>72.1 71.8 70.3</td>
</tr>
<tr>
<td>GSI</td>
<td>83.2 83.1 82.3</td>
<td>68.8 67.8 66.0</td>
</tr>
<tr>
<td>GMC+SI</td>
<td>85.2 85.1 83.9</td>
<td>67.8 69.1 65.8</td>
</tr>
<tr>
<td>HNC</td>
<td>70.3 64.0 64.1</td>
<td>65.8 61.8 60.4</td>
</tr>
<tr>
<td>LABP D</td>
<td>81.0 78.9 78.2</td>
<td>77.1 74.5 73.7</td>
</tr>
<tr>
<td>LABP C</td>
<td>81.4 81.6 80.7</td>
<td>76.9 76.7 76.8</td>
</tr>
<tr>
<td>LNBPOA</td>
<td>79.7 79.8 78.7</td>
<td>76.3 76.0 75.1</td>
</tr>
<tr>
<td>LNP TA1</td>
<td>82.4 82.6 82.2</td>
<td>77.9 77.9 77.9</td>
</tr>
<tr>
<td>LNP TA2</td>
<td>79.3 79.5 78.2</td>
<td>75.7 75.4 74.7</td>
</tr>
<tr>
<td>LABP D+LNBPOA</td>
<td>82.9 82.4 82.0</td>
<td>79.6 78.8 78.4</td>
</tr>
<tr>
<td>LDBP+LABP D</td>
<td>87.0 86.4 86.3</td>
<td>84.2 83.7 83.7</td>
</tr>
<tr>
<td>LDBP+LABP C</td>
<td>86.4 86.9 86.9</td>
<td>84.0 84.3 84.8</td>
</tr>
<tr>
<td>LDBP+LNBP OA</td>
<td>87.4 86.8 86.6</td>
<td>85.4 84.8 84.5</td>
</tr>
<tr>
<td>LDBP+LNBP TA2</td>
<td>86.3 85.9 85.3</td>
<td>84.1 83.5 83.1</td>
</tr>
<tr>
<td>LDBP+LNBP TA1</td>
<td>86.9 87.0 86.9</td>
<td>84.6 84.6 84.7</td>
</tr>
<tr>
<td>LDPQ</td>
<td>88.3 88.0 86.6</td>
<td>86.1 85.9 84.6</td>
</tr>
<tr>
<td>LAPQ</td>
<td>79.7 78.8 78.2</td>
<td>76.3 74.9 75.0</td>
</tr>
<tr>
<td>LDGBP</td>
<td><strong>89.9</strong> 89.4 <strong>88.8</strong></td>
<td><strong>88.3</strong> 87.8 <strong>87.1</strong></td>
</tr>
<tr>
<td>LAGBP</td>
<td>83.0 82.1 81.7</td>
<td>80.0 79.4 78.9</td>
</tr>
<tr>
<td>LDMPB</td>
<td>89.8 <strong>89.5</strong> 88.7</td>
<td>88.1 <strong>88.0</strong> 86.8</td>
</tr>
<tr>
<td>LAMB P</td>
<td>82.1 82.1 80.2</td>
<td>79.1 79.0 76.9</td>
</tr>
</tbody>
</table>

In order to further examine the performance of the different features, cross-database testing was conducted. Here the Bosphorus database was used to train the classifiers, and the D3DFACS database [35] was used for testing. As it is not known precisely where the AUs start and end within the sequences in this database, two tests were performed: firstly only on the middle half of the sequence, i.e. the apex and high intensity frames, and then secondly on all the frames in the sequences. Due to the frontal 2D images not being available for this database, it was not possible to perform cross-database tests using the 2D features for comparison. In order to determine a single set of confidence values for the sequence, all testing frames were tested individually, and then the confidence levels across the frames were averaged and the the ROC AuCs calculated. The results can be...
seen in Table 3.8. This method was shown to produce higher results than taking the individual frame rates alone, as it allowed smoothing which reduced the effect of frames that were difficult to classify. Three different training sets were compared, consisting of different ranges of intensity examples from the Bosphorus database, as shown in the three columns. As would be expected, the results when testing on all frames were generally lower than only on the high intensity frames. In addition, the results showed that training on the full set of intensities did reduce performance in both cases, though less so when testing on all frames, as would be expected. However, using the B-E training set did not greatly affect performance.

The results show interesting behaviour in the comparative performance of the features. The LDGBP and LDMBPs continue to produce the highest average results, scoring 89.9 and 89.8 AuC respectively when trained on the C-E examples. However, this time we see that the majority of normal and azimuthal methods yield significantly lower results than the depth-based features. The reduced performance of the angle based features suggests that they are more susceptible to the high frequency variations between the two databases, introduced by the large differences in smoothness in the data. This is because the normal direction, and thus azimuthal angle, can be greatly affected by small amounts of noise in the mesh. The application of the Gabor and Monogenic methods to the APDI does still show improvement over the basic LABP or LNBP methods, suggesting that filtering is still able to extract more useful information, as it can remove some of this high frequency noise. In addition, the full results (included in the supplementary material) show that the individual AU results show that there are still some actions which the azimuthal or normal-based features are better able to distinguish, and combining these features still results in some improvement in this cross-database test. However, generally, the filtered normal-based features still fail to achieve results that are comparable to their depth counterparts.

### 3.6.6 Summary and Discussion

In these experiments the effectiveness of applying binary pattern algorithms to 3D facial geometries for the purpose of AU detection was demonstrated. The full performance re-
3.6. Experiments

Results show there is no one feature that is superior for this task. Normal-based analysis improves detection performance over 2D intensity image and 3D depth map binary patterns for a number of AUs, though they perform comparably well on average, and are outperformed for some AUs. However, the application of filtering techniques prior to binary analysis gives a significant improvement in performance, particular for depth based LDPQs, LDGBP and LDMBP, and azimuthal LAGBP features. However, because of the ability for LDMBP to achieve comparable performance to the LDGBP, but with a much shorter feature descriptor length, a key finding of this Chapter is that they are particularly useful for analysis of AUs in 3D data.

However, all features that were explored performed well on some of the AUs in the set. This demonstrates that, rather than being able to choose a single feature as superior for AU detection, it is the combination of a number of feature types that is best suited to the task of detection of the full set of AUs possible in the face. It was demonstrated that combining each of the basic features with 2D features or LDGBP can give significant improvements in performance, and even achieve results comparative to those attained through filtering techniques. Similarly, combining depth based and normal based filtered features with 2DLGBP also gives significant increases in performance over the single features alone.

Exploring the optimal parameters for each feature type demonstrated that for the majority the radius size did not affect performance hugely. However, a radius of 8 or 11 appeared to be best for most feature types, with only the LAPQ appearing to require a much smaller region size of 3. The low intensity experiments showed how the proposed features achieve state of the art results when combined with 2D features, particularly LDGBP and LDMBP. This performance demonstrates the robust nature of the feature combinations, even when detecting very subtle motions in the face.

The cross-database experiments demonstrated that how depth based features are able to generalise well to new data, achieving high performance on a new database of AU examples. In contrast, though the normal based approaches are able to capture useful information about the AUs above that of the depth alone, their performance suffers due to their sensitivity to different noise levels in the two databases, so they do not generalise
3. Binary Pattern Features for 3D Facial Action Unit Detection

<table>
<thead>
<tr>
<th>AU</th>
<th>Image</th>
<th>Mesh</th>
<th>LDBP</th>
<th>LABP&lt;sub&gt;D&lt;/sub&gt;</th>
<th>LNBP&lt;sub&gt;TA&lt;/sub&gt;</th>
<th>LNBP&lt;sub&gt;OA&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 mouth corner dimples</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="mesh1" alt="Mesh" /></td>
<td><img src="ldbp1" alt="LDBP" /></td>
<td>![LABP]&lt;sub&gt;D&lt;/sub&gt;(1)</td>
<td>![LNBP]&lt;sub&gt;TA&lt;/sub&gt;(1)</td>
<td>![LNBP]&lt;sub&gt;OA&lt;/sub&gt;(1)</td>
</tr>
<tr>
<td>15  (\sim) shape of mouth</td>
<td><img src="image2" alt="Image" /></td>
<td><img src="mesh2" alt="Mesh" /></td>
<td><img src="ldbp2" alt="LDBP" /></td>
<td>![LABP]&lt;sub&gt;D&lt;/sub&gt;(2)</td>
<td>![LNBP]&lt;sub&gt;TA&lt;/sub&gt;(2)</td>
<td>![LNBP]&lt;sub&gt;OA&lt;/sub&gt;(2)</td>
</tr>
<tr>
<td>16 bottom lip extrudes</td>
<td><img src="image3" alt="Image" /></td>
<td><img src="mesh3" alt="Mesh" /></td>
<td><img src="ldbp3" alt="LDBP" /></td>
<td>![LABP]&lt;sub&gt;D&lt;/sub&gt;(3)</td>
<td>![LNBP]&lt;sub&gt;TA&lt;/sub&gt;(3)</td>
<td>![LNBP]&lt;sub&gt;OA&lt;/sub&gt;(3)</td>
</tr>
<tr>
<td>20 lips stretched slight</td>
<td><img src="image4" alt="Image" /></td>
<td><img src="mesh4" alt="Mesh" /></td>
<td><img src="ldbp4" alt="LDBP" /></td>
<td>![LABP]&lt;sub&gt;D&lt;/sub&gt;(4)</td>
<td>![LNBP]&lt;sub&gt;TA&lt;/sub&gt;(4)</td>
<td>![LNBP&gt;&lt;sub&gt;OA&lt;/sub&gt;(4)</td>
</tr>
<tr>
<td>23 lips tensed thinned mouth</td>
<td><img src="image5" alt="Image" /></td>
<td><img src="mesh5" alt="Mesh" /></td>
<td><img src="ldbp5" alt="LDBP" /></td>
<td>![LABP]&lt;sub&gt;D&lt;/sub&gt;(5)</td>
<td>![LNBP]&lt;sub&gt;TA&lt;/sub&gt;(5)</td>
<td>![LNBP&gt;&lt;sub&gt;OA&lt;/sub&gt;(5)</td>
</tr>
<tr>
<td>28 lips turned inwards</td>
<td><img src="image6" alt="Image" /></td>
<td><img src="mesh6" alt="Mesh" /></td>
<td><img src="ldbp6" alt="LDBP" /></td>
<td>![LABP]&lt;sub&gt;D&lt;/sub&gt;(6)</td>
<td>![LNBP&gt;&lt;sub&gt;TA&lt;/sub&gt;(6)</td>
<td>![LNBP&gt;&lt;sub&gt;OA&lt;/sub&gt;(6)</td>
</tr>
</tbody>
</table>

Figure 3.13: Comparing the information captured by the different feature types for a number of AU examples. Features highlighted in dark grey achieve the highest performance within their category (basic or filter-based), and light grey features also perform well.

as well. This does not render these features useless however, as for applications where the train and test data will be captured using the same system the azimuthal features will still be useful for analysis of a number of AUs, and sometimes outperform the depth based features. However, for very different data this shows that depth based features will be required, and the combination of both types of features can still give improved performance overall.

By looking more closely at the shapes formed when different AUs are displayed, the benefits of each feature type becomes clearer. Tables 3.13 and 3.14 compare the information
3.6. Experiments

<table>
<thead>
<tr>
<th>AU</th>
<th>LDPQ</th>
<th>LDGBP</th>
<th>LAGBP</th>
<th>LDMBP</th>
<th>LAMBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 mouth corner</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>dimples</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>15 ~ shape of mouth</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>16 bottom lip</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>extrudes</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>20 lips stretched</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>slight ~</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>thinned mouth</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>28 lips turned</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>inwards</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

Figure 3.14: Comparing the information captured by the different filter-based feature types for a number of AU examples. Features highlighted in dark grey achieve the highest performance, and light grey features also perform well.

captured by the basic and filter-based features respectively, for a number of AU examples, in order to see what in particular each capture. Each of the AUs in this table has particular distinguishing characteristics, as shown in the first column. The remaining columns in this table then display how some features are able to pick out characteristic shapes, explaining why these features achieve the highest performance for these AUs. For example, this table shows how LABPs and LDPQs are both particularly good at picking out the important ~ shape that the mouth makes that is characteristic of AU 15, as well as the subtle wrinkling that is present at the corners of the mouth in this AU. This is far more visible in these feature images than the others shown here. This is also an example of an
3. Binary Pattern Features for 3D Facial Action Unit Detection

AU where the LAGBP outperforms the LDGBP feature type, and it is clear why this is from these images - the phase information captured by the LDGBP does not capture the important \(\sim\) shape, nor show any mouth corner wrinkling, whereas these are visible in the LAGBP image. These differences are reflected in the superior performance scores of these features, seen in Table 2. This table also shows how AU 20 is better distinguished by the azimuthal-based features than the depth-based features. The important characteristic of this action unit (the stretching outwards, but also slightly downwards, of the mouth corners) causes the mouth to appear thicker than in other AUs in which the mouth stretches outwards, such as AUs 12 and 14, and these examples show how this aspect is particularly well captured by the LABP and LAGBP features. This is also reflected in the performance scores. The remaining rows in Tables 3.13 and 3.14 also show similar examples of how other features are able to capture the characteristic shapes important for AUs 14, 16, 23 and 28.

Finally, the results presented here show the comparative performance of 2D and 3D features in controlled conditions, with consistent illumination and no head pose changes. However, in real video data, the images will be affected by variations in the lighting, and changes in head position. It is therefore expected that the performance of 2D features will be reduced significantly by lighting (as this affects the intensity appearance) and that both 2D and depth-based features will be affected by head position changes, as they are not rotation invariant. However, it is on data of this kind where it would be expected that, as well as adding useful extra information about the AU motion, 3D features, and particularly azimuthal and normal-based features, would prove to be comparatively robust, as they are unaffected by lighting, and pose changes can be removed easily through alignment of facial meshes. In addition, the normal-based approaches are rotation invariant, provided that the feature regions remain the same under the rotation applied. For this reason it is expected that, under these conditions, the new proposed features would greatly aid AU detection.
3.7 Conclusions

In this chapter, the performance of binary pattern based features for the task of 3D facial AU detection has been fully explored. For this purpose a number of representations of the 3D data have been exploited, the depth map, facial normals, APDI and APCIs, to create a new set of feature types based on the ideas of LBP, LPQs, and Gabor and Monogenic methods, that have been successfully applied in the 2D case. Classifier level fusion of these different modalities was explored, and shown to increase accuracy. GB feature selection, including a new region selection approach, and SVM classification, were applied in order to test the comparative performance of the features in cross-validation on the Bosphorus database. Comparisons were conducted against a number of alternative features, and fusion with 2D features was also explored for increased performance.

The generalisability of the features to new data was then examined, through cross-database testing using the D3DFACS database. These results demonstrated that the 3D features were largely complementary to the original 2D operators, and that, in particular, the LDGBP and LDMBP feature types were able to achieve very high performance. The azimuthal feature types also showed promise in the cross-validation tests, however they struggled to generalise as well in the cross-database test, as they are more sensitive to large changes in smoothness between the datasets.
Chapter 4

Dynamic 3D Facial Expression Analysis

*This work was published as [SZPR12], with preliminary results published in [SZPR11].*

4.1 Introduction

Expression dynamics are of great importance for the interpretation of human facial behaviour [113]. They are able to convey vital cues for behaviour interpretation not possible from static data alone [3], and are crucial for distinguishing between spontaneous and posed emotional expressions [178]. In addition, they are essential for the recognition of complex states such as pain and mood, [192], as well as of more subtle emotions such as social inhibition, embarrassment, amusement and shame [36, 47]. Harnessing the information available in expression dynamics is therefore a necessary part of any system capable of accurate and robust expression recognition.

Expression dynamics and 3D facial geometry data can be combined to offer a wealth of information that can be employed for the analysis of facial expressions. The development of such systems will open up new avenues in facial expression recognition as, unlike 2D data, 3D facial geometries ensure that all motion in the face is captured, and analysis of full expression dynamics allow cues to be detected that are unavailable in static data.

Expression sequences can be broken down into four temporal segments: neutral - absence of the expression, onset - when the expression is increasing in strength, apex - when the expression intensity does not change, and offset - when the expression is decreasing in
strength. This chapter proposes a method that aims to exploit these temporal segments in 4D data through the extraction of 3D motion-based features and temporal modelling of the full expression dynamics for recognition purposes.

A fully automatic method for analysis of expression dynamics is proposed which consists of several stages. Firstly, the 3D motion of the face appearing between frames in each image sequence is captured using FFDs [134]. Features are extracted by applying a quad-tree decomposition of the motion fields. Features selection is conducted via a GB method for each of the onset and offset temporal segments of the expression, and the same method is applied for training classifiers of each segment. Temporal modelling of the full expression is then performed through HMMs which model the full expression dynamics (neutral-onset-apex-offset). These models are then employed for dynamic expression recognition. A comparison is conducted this method and the equivalent 2D system which using a similar methodology. These experiments prove the superiority of the 3D approach.

In summary, the contributions of this chapter are as follows:

- Employing 3D FFDs, sets of 2D vector projections and quad-trees in order to perform 3D motion-based feature extraction.

- An extension of the method proposed in [75] to perform expression recognition using both 2D intensity images and 3D facial geometry information.

- A comparison of the equivalent 2D and 3D methods is then performed on the same database in order to assess the benefits of the 3D data.

This chapter is organised as follows: Section 4.2 outlines the motion-based features employed, and the method for extraction of these. Section 4.3 details the classification and temporal modelling method employed to harness the full expression dynamics. Section 4.4 presents the results of the experiments conducted to test the method, and finally Section 4.5 concludes the chapter.
4.2 Feature Extraction

Here the full feature extraction method is described. The features employed are based on the motion between frames, and the method for extraction of the features consists of several stages. Firstly, vector fields are captured via FFDs [135]. Vector projections are computed for each pair of axes \((x, y, z)\) and \(t\), in a similar manner to the method exploited in [75]. However, here it is necessary to compute projections for all three spatial dimensions, resulting in six different projections. These are then employed to focus interest only on the areas in which the greatest amount of motion occurs via quadtree decompositions, which are applied on the vector projections to divide the vector field into regions according to the amount of motion in every region. Finally, a set of features are extracted from each region.

4.2.1 Motion Extraction

The motion between the frames in each image sequence is captured using 3D FFDs. FFDs [135] is a method for non-rigid registration based on B-spline interpolation between a lattice of control points. The 2D version of this method is employed for motion capture in [75]. The aim is given two meshes, with vertices \(p = (x, y, z)\) and \(\dot{p} = (\dot{x}, \dot{y}, \dot{z})\) respectively, to find a vector field given by \(T(p)\) such that:

\[
\dot{p} = T(p) + p. \tag{4.1}
\]

The basic idea is to deform the target object by manipulating an underlying mesh of control points. The lattice, \(\Phi\), is regular in the source image and consists of \(n_x \times n_y \times n_z\) points \(\phi(i, j, k)\) with regular spacing. It is then deformed by registration of the points in the target image to become \(\Phi'\) with irregularly spaced control points. The difference between the two lattices is denoted as \(\Phi_\delta\). \(T(p)\) can be computed using B-spline interpolation on \(\Phi_\delta\).

For any point in the 3D mesh \(p\), let the closest control point have coordinates \((x_0, y_0, z_0)\) and displacement \(\phi_\delta(i, j, k)\). The transformation of this point can be given as the B-spline
4. Dynamic 3D Facial Expression Analysis

Algorithm 3 Non-rigid registration algorithm

Require: Stopping criterion $\epsilon$

Require: Step size $\mu$

Initialise the control points $\Phi'$

Calculate the gradient vector of the cost function $C$ with respect to the current control points $\Phi'$:

$$\nabla C = \frac{\partial C(\Phi')}{\partial \Phi'}$$

while $||\nabla C|| > \epsilon$ do

Recalculate the control points

$$\Phi' = \Phi' + \mu \frac{\nabla C}{||\nabla C||}$$

Recalculate the gradient vector $\nabla C$

end while

Calculate $\Phi_\delta = \Phi - \Phi'$

Derive $T(p)$ through B-spline interpolation of $\Phi_\delta$

interpolation of the 64 closest control points:

$$T(p) = \sum_{i=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_l(a_1) B_m(a_2) B_n(a_3) \phi_\delta(i + l, j + m, k + n)$$

where $a_1 = x - x_0$, $a_2 = y - y_0$, $a_3 = z - z_0$, and $B_l$ is the $l^{th}$ basis function of uniform cubic B-spline, defined as follows:

$$B_0(a) = \frac{1}{6}(-a^3 + 3a^2 - 3a + 1)$$

$$B_1(a) = \frac{1}{6}(3a^3 + 6a^2 + 4)$$

$$B_2(a) = \frac{1}{6}(-3a^3 + 3a^2 + 3a + 1)$$

$$B_3(a) = \frac{1}{6}a^3.$$  

$T(p) = (u(p), v(p), w(p))$ is the vector field employed in this work for expression analysis.

In order to calculate $\Phi_\delta$, a cost function $C$ is defined. Here $C$ is chosen to be the sum of squared differences between the points in the target and reference meshes. The non-rigid registration algorithm then proceeds to optimise the control point lattice, $\Phi'$, by minimising this cost function. To do this, an iterative gradient descent technique is employed with step size $\mu$ in the direction of the gradient vector. The algorithm finishes
4.2. Feature Extraction

Figure 4.1: Example of 2D FFDs applied to aligned face images. Grid shows the control point lattice and the B-spline interpolation of this. (a) Start of onset of Smile (b) End of onset of Smile

when a local optimum is found, which in this case is defined as when the gradient of the cost function reaches a suitably small positive value. The difference between the optimised control point lattice and the original regular lattice is then calculated, and this is exploited to perform B-spline interpolation in order to find the vector field that captures the motion between the frames. The full algorithm is shown in Algorithm 3.

Fig. 4.1 shows an example of applying 2D FFDs to extract the motion between a pair of images displaying a smile. Here the lattice of control points and the B-spline interpolation between them is shown as a yellow grid. In Fig. 4.1a the grid is almost regular, whereas in Fig. 4.1b this grid has been deformed in order to capture the bulging of the cheeks and stretched lips around the mouth.

The resolution of the grid used determines the sensitivity of finely motion tracking between the two images. In this work a grid with control point spacing of \(1\text{mm}\) is employed. Fig. 4.2 shows neutral and apex 3D facial geometry meshes for an example of the happiness expression, and the motion tracked by FFDs between these frames. The most highly concentrated areas of motion are around the corners of the mouth and the cheeks, as is expected for this expression.
4. Dynamic 3D Facial Expression Analysis

Figure 4.2: Mesh representations of neutral and apex frames taken from the Happy image sequence for subject F004, along with the motion tracked between them by FFDs. (a) Mesh of the cropped neutral 3D facial geometry (b) Mesh of the cropped apex 3D facial geometry (c) Vector field showing the motion between these frames

4.2.2 Vector Projections

Vector projections, displayed as an image, show the areas in the image in which there is a high concentration of motion in the sequences across a number of frames or spatial axis. Two sets of vector projections are produced from the dataset, one built from frames in which the onset segment of the expression occurred, and other from frames in which the offset segment of the expression occurred. Six 2D vector projections are created from the 3D facial motion. These consist of three spatial vector projections, one for each pair of spatial axes, and three time-space vector projections.

For each $i^{th}$ sequence, the set of frames belonging to the temporal sequence (either onset or offset) is defined as $\Omega_i$. A sliding window is then defined as centred on $\tau$ with width $\theta$, where $\tau \in \Omega_i$. The spatial vector projections for a window width of $\theta$ is then calculated by summing the absolute motion at a particular coordinate in two dimensions (e.g. $(x, y)$) across one dimension (e.g. across $z$ for projection $P^\theta_{xy}$) across each sliding window in all sequences. Hence:

$$P^\theta_{xy}(x, y) = \sum_{i=1}^{M} \sum_{\tau \in \Omega_i} \sum_{t=\tau-\theta}^{\tau+\theta+1} \sum_{z} u^2_{i,x,y,z,t} + v^2_{i,x,y,z,t} + w^2_{i,x,y,z,t} \quad (4.3)$$
4.2. Feature Extraction

Figure 4.3: Spatial and space-time vector projections and the quad-trees they produced for the onset segment of the Happy expression with window width of 4. Spatial vector projections (a) $x - y$ (b) $x - z$ (c) $y - z$, spatial quad-trees (d) $x - y$ (e) $x - z$ (f) $y - z$, space-time vector projections (g) $x - t$ (h) $y - t$ (i) $z - t$, space-time quad-trees (j) $x - t$ (k) $y - t$ (l) $z - t$. 

107
where $M$ is the total number of image sequences of the current expression in the training set, and

$$u_{i,x,y,z,t} = u^i(x, y, z, t),$$
$$v_{i,x,y,z,t} = v^i(x, y, z, t),$$
$$w_{i,x,y,z,t} = w^i(x, y, z, t)$$

are the vector components, in the $x$, $y$ and $z$ directions respectively, at coordinates $(x, y, z)$ and time $t$ in the $i^{th}$ image sequence. Note the summation is performed over the window to be used, as well as over the sequence, to ensure all frames employed for gathering features influence the quad-tree decomposition.

The time-space vector projections are calculated for the width of the sliding windows, hence for $t$ values in the range $0 \leq t \leq 2\theta - 1$. They are calculated from summing the motion in the component in question across two dimensions, across all frames in the temporal segment for all sequences. Hence:

$$P_{x}^{\theta}(x, t) = \sum_{i=1}^{M} \sum_{\tau \in \Omega_i} \sum_{t=\tau - \theta}^{\tau + \theta + 1} u_{i,x,y,z,t}^2 + v_{i,x,y,z,t}^2 + w_{i,x,y,z,t}^2$$

(4.6)

$$P_{y}^{\theta}(y, t) = \sum_{i=1}^{M} \sum_{\tau \in \Omega_i} \sum_{t=\tau - \theta}^{\tau + \theta + 1} u_{i,x,y,z,t}^2 + v_{i,x,y,z,t}^2 + w_{i,x,y,z,t}^2$$

(4.7)

$$P_{z}^{\theta}(z, t) = \sum_{i=1}^{M} \sum_{\tau \in \Omega_i} \sum_{t=\tau - \theta}^{\tau + \theta + 1} u_{i,x,y,z,t}^2 + v_{i,x,y,z,t}^2 + w_{i,x,y,z,t}^2$$

(4.8)

Examples of vector projections can be seen in Figs. 4.3a-4.3c and Figs. 4.3g-4.3i, here collected from one fold of onset of the Happy expression with window width of 4. The former shows the spatial vector projections and the latter the space-time vector projections.
4.2. Feature Extraction

Algorithm 4 Quad-tree decomposition

Require: Splitting threshold $\gamma$
Require: Minimum region size $\sigma$
Require: Set of regions $R$

Define $p_{tot}$ as the total sum of movement across the full image
Initialise $R$ with single region which is entire image

while True do
  for region $r$ in $R$ do
    Set $p$ to be sum of movement in $r$
    if $p > \gamma p_{tot}$ and size of $r > \sigma$ then
      remove $r$ from $R$
      divide $r$ into four equally sized square regions
      add these new regions to $R$
    end if
  end for
  if no region is divided then
    Stop
  end if
end while

4.2.3 Quad-Tree Decomposition

Before feature extraction could be performed on each of the image sequences, the images are divided into regions from which a set of features are acquired. Instead of dividing the images into evenly sized regions, the technique employed is quad-tree decomposition. Quad-tree decomposition has been widely exploited in computer vision and image processing for image segmentation and feature extraction. In this case quad-tree decompositions are employed to divide the image into regions sized according to the amount of motion present in each part of the vector projection. The algorithm, detailed in Algorithm 4, works by measuring the percentage of total motion in the frame that is contained in each region. A region is divided into four equally sized square regions if the percentage it contains is over a certain threshold. A lower limit is set on the region size, below which the regions cannot be divided further. The division continues repeatedly until no further regions can be split. The threshold, $\gamma$, used is 6\% of the average amount of motion in the blocks. This was determined to give adequate quad-tree decomposition results from preliminary testing. Two sets of quad-tree decompositions are found from the training set - one from the frames consisting of onset motion, and one from frames consisting of offset motion. These sets are then employed throughout the training and testing.
4. Dynamic 3D Facial Expression Analysis

Sliding windows are employed throughout the quad-tree decomposition and feature extraction in order to allow information from previous or later frames to be exploited in the classification of the current frame. This is useful as the duration of a certain motion can help with differentiating between two or more expressions. Various window widths are tested to identify which width gave the best results for each expression. Here window width refers to the distance from the central frame to the outermost frame in the window. Hence, a window width of $\theta$ will mean that $2\theta$ frames have influence on the central frame.

Examples of the quad-trees produced for each of the vector projections in Fig. 4.3 can be seen in Figs. 4.3d-4.3f and Figs. 4.3j-4.3l. For example, Fig. 4.3e shows the decomposition created by dividing the vector projection in Fig. 4.3b according to the amount of motion in the image. The smallest regions correspond to those parts of the image that contain the highest concentration of the motion, whereas the larger regions contain very little motion.

### 4.2.4 Features

Once the quad-trees had been produced for each of the vector projections they are employed to extract features for every frame in the set of image sequences. For each region in the quad-tree, one set of 3D features are identified and stored. Therefore, areas where little motion is observed will be covered by large regions and so produce few features, whereas areas with a large amount of motion produced small regions and so gave many features. The features exploited included the mean and standard deviation of the distribution of directions of the vectors in that region, the magnitude of the total motion, and the divergence and curl of the vector field in the region. The features from all the regions are concatenated into one feature vector per frame in the image sequences, and these are utilised for classification.

Again, a sliding window is employed to allow frames before or after the current frame to influence the features gathered for that frame. Hence, the features are extracted for a window of width $\theta$ around the current frame which is at time $\tau$ in the image sequence. The vector field for the frames in this window are averaged across either space or time using a similar calculation to that employed for the vector projections. The quad-trees previously
4.3 Modelling Expression Dynamics

In this section, the full process for expression modelling is described. The features are extracted from the motion between frames in the full sequence, as described in the previous section. They are then gathered from each region in each frame, and are employed to train classifiers on the onset and offset segments of the expression. The outputs are exploited to build a HMM of the full expression sequence. An overview of the system can be seen in Fig. 4.4.

4.3.1 Classification

Once features for a set of image sequences are extracted, the next step is application of GB classifiers, in order to simultaneously select the best features to employ, and perform
classification training. Two classifiers are trained for each expression: one for each of the onset and offset temporal segments.

Target labels are created for each classifier by setting the labels for frames belonging to the temporal segment to be 1, and all other frames to be −1. These are exploited, along with the features matrix produced from each set of quad-trees, as input to the classifiers. At each iteration in the training algorithm, the classifier chooses a feature that reduces the error by the largest margin, and then stores this feature and the associated parameters. This continues until the error rate no longer reduces, or the maximum number of features is reached, here set to be 200.

Once the two classifiers have been fully trained they are employed to test the same set of
4.3. Modelling Expression Dynamics

Figure 4.6: The HMM transition model consisting of neutral, onset, apex and offset states and the transitions possible between them.

features. This produces a set of predicted labels for the frames in the training set, along with confidence levels for these labels. The labels and confidences are multiplied together to form a distribution of values suitable as input to the HMMs.

Testing is then conducted against each of these classifiers in order to produce emission values. An example of the output from the two classifiers throughout a test Sad sequence can be seen in Fig. 4.5. Values above the zero \( x \)-axis indicate frames which are labelled as belonging to the corresponding segment, onset or offset.

### 4.3.2 Temporal Modelling

HMMs are employed to model the temporal dynamics of the entire expression. These are trained on the emission output from the GB classifiers which is formed by taking the labels and confidence values from testing of the training set and multiplying these together.
A sequence displaying a full expression is modelled as containing four temporal segments - neutral, onset, apex and offset. These form the basis for four possible states of the hidden variable in the HMM. The general form of the model for one expression can be seen in Fig. 4.6. This model allows transitions from each state to the next, as well as to itself, but also from apex back to onset, and from offset back to apex, to reflect the fact that for some expressions the subject can have multiple apexes. The actual transition probabilities for each expression are calculated from the labels in the training set, and so the latter two transitions may not be possible for all expressions if examples do not appear in the training set. The model assumes that the expression will start in neutral or onset, progressing through all of the other three states, until finally returning to neutral. Hence only frames at the beginning and end of the sequence are labelled as neutral, and all other stationary frames in between are labelled as apexes. This is appropriate in these experiments, as the examples from the BU-4DFE utilised all contain sequences of this form.

The three sets of parameters of an HMM are:

- **Initial Probabilities** - the probability distribution of the initial states across the image sequences.

- **Transition Probabilities** - a matrix defining the probabilities of the different transitions between underlying states in the model.

- **Emission Probabilities** - the conditional probability distribution defining how the observed values depend on the hidden states.

Each of these is determined by the results gathered from testing the trained classifiers. Let \( L \) be a matrix containing the state labels for the training set of frames, where each row corresponds to a different image sequence, and each column to a different frame index in this sequence. In practise this is stored as an array of cells as the image sequences are of different lengths and so contain different numbers of frames. In addition, let \( E^{on} \) and \( E^{off} \) be matrices containing the emission values produced by the onset and offset classifiers respectively. The initial probability distribution, \( P \), is calculated by estimating
the prior probabilities from the state labels of the first frame in each image sequence in the training set. The transition probability matrix, $T$, is also estimated from the state labels by using the frequency of each transition between states.

Finally the emission probability distribution must be calculated using the emission values and the labels. Gaussian distributions are employed, for which the mean, $\mu$, and standard deviation, $\sigma$, of the possible emission values for each of the possible states are determined. Hence the distribution is represented by two matrices each with four rows corresponding to the four states, and two columns corresponding to the two classifiers, onset and offset. The mean matrix, $M$, is calculated by averaging the emission values observed for each of the temporal states:

$$M_{(1,s)} = \frac{1}{N_s} \sum_{(i,j) \in f(s)} E_{on}^{(i,j)},$$
$$M_{(2,s)} = \frac{1}{N_s} \sum_{(i,j) \in f(s)} E_{off}^{(i,j)},$$

where $N_s$ is the total number of frames in $L$ with label $s$, and

$$f(s) = \{(i,j)|L_{(i,j)} = s\}.$$

The standard deviation matrix, $S$, can be calculated as:

$$S_{(1,s)} = \sqrt{\frac{1}{N_s} \sum_{(i,j) \in f(s)} (E_{on}^{(i,j)} - M_{(1,s)})^2},$$
$$S_{(2,s)} = \sqrt{\frac{1}{N_s} \sum_{(i,j) \in f(s)} (E_{off}^{(i,j)} - M_{(2,s)})^2}.$$

Once these properties of the HMM had been estimated from the training data, the model can be employed for testing new image sequences. Testing is conducted by collecting features from the new image sequence using the same quad-trees created from the training set, testing the classifiers on these features, and then using the observed values along with the standard Viterbi algorithm to determine the most likely sequence of states.

### 4.4 Experiments

Experiments were conducted on the BU-4DFE database [205]. The image sequences available in the database were filtered to remove any expressions that were deemed to be
4. Dynamic 3D Facial Expression Analysis

Figure 4.7: Validation $F_1$-measures for each expression for all window widths (a) Happy (b) Sad (c) Angry (d) Disgust (e) Surprise (f) Fear

inaccurate representations, or any sequences that did not end in the neutral expression. This resulted in the following numbers of examples available to utilise for each expression: Happy - 90, Sad - 58, Angry - 53, Disgust - 59, Surprise - 87 and Fear - 50.

The testing was done using 6-fold cross-validation. For each fold to be tested, a training set was created for each expression from the other 5 folds. The method employed for construction of a suitable training set involved taking all available positive examples from the folds, and then an equal number of negative examples by randomly selecting from the remaining expressions available. These training sets were employed to train one classifier to model each expression.

The next stage in the testing was choosing a suitable window width for each expression. This was done by performing a validation test for each of the expressions and window widths. This test looked at the ability of each classifier to discriminate between positive and negative examples of the expression for which it has been trained. The window width which gave the best validation $F_1$-measure could then be chosen as the most suitable
4.4. Experiments

Table 4.1: \(F_1\)-measures achieved with 2D and 3D testing. WW = Window width, RR = Recall rate, PR = Precision rate, \(F_1 = F_1\)-measure.

<table>
<thead>
<tr>
<th>Expression</th>
<th>2D System</th>
<th></th>
<th>3D System</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WW</td>
<td>RR</td>
<td>PR</td>
<td>(F_1)</td>
<td>WW</td>
<td>RR</td>
<td>PR</td>
<td>(F_1)</td>
<td></td>
</tr>
<tr>
<td>Happy</td>
<td>8</td>
<td>76.40</td>
<td>77.27</td>
<td>76.84</td>
<td>16</td>
<td>71.91</td>
<td>84.21</td>
<td>77.58</td>
<td></td>
</tr>
<tr>
<td>Sad</td>
<td>12</td>
<td>44.83</td>
<td>42.62</td>
<td>43.70</td>
<td>12</td>
<td>70.69</td>
<td>57.75</td>
<td>63.57</td>
<td></td>
</tr>
<tr>
<td>Angry</td>
<td>12</td>
<td>61.54</td>
<td>58.18</td>
<td>\textbf{59.81}</td>
<td>12</td>
<td>48.08</td>
<td>45.45</td>
<td>46.73</td>
<td></td>
</tr>
<tr>
<td>Disgust</td>
<td>12</td>
<td>57.63</td>
<td>65.38</td>
<td>\textbf{61.26}</td>
<td>16</td>
<td>54.24</td>
<td>52.46</td>
<td>53.33</td>
<td></td>
</tr>
<tr>
<td>Surprise</td>
<td>12</td>
<td>79.07</td>
<td>78.16</td>
<td>78.61</td>
<td>12</td>
<td>79.07</td>
<td>85.00</td>
<td>\textbf{81.93}</td>
<td></td>
</tr>
<tr>
<td>Fear</td>
<td>8</td>
<td>38.46</td>
<td>37.50</td>
<td>37.97</td>
<td>8</td>
<td>43.59</td>
<td>42.50</td>
<td>\textbf{43.04}</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>59.65</td>
<td>59.85</td>
<td>59.70</td>
<td></td>
<td>61.26</td>
<td>61.23</td>
<td>\textbf{61.03}</td>
<td></td>
</tr>
</tbody>
</table>

window width to employ in the six-way expression decision. The validation \(F_1\)-measures can be seen in Fig. 4.7 for each of the expressions. Initially, the window width chosen for each expression was automatically selected as being the one that produced the highest \(F_1\)-measure in the validation tests. Then, for expressions that had more than one window width that gave a similar \(F_1\)-measure, an alternative window width was manually chosen if it improved the results of the six-way classification process.

Then each of the sequences in the test set was tested against all six of the classifiers, and the outputs exploited to make a decision about which expression the sequence represented. The method for determining the predicted expression was as follows. Firstly, only the sequences for which one or more frames were labelled as the apex state were considered. Finally the most appropriate expression label was chosen by taking the expression for which the sequence containing the apex was most likely.

An additional test was run in order to measure the benefit of using 3D facial geometries
4. Dynamic 3D Facial Expression Analysis

over 2D image sequences for facial expression recognition. Here the 2D facial intensities available from the BU-4DFE were employed along with an adapted version of the system proposed in [75] which could model the full expression and make a six-way decision in a similar way to the 3D method. The differences in the 2D tests as compared to 3D were: the alignment used between image sequences required manual eye detection as opposed to that used with the 3D method which was fully automatic. 2D FFDs were exploited to compute the motion between frames in each sequence. For feature extraction and classification similar methods were employed as in [75], and then HMMs were employed to model the full expression as done for 3D. Hence a comparison between 2D and 3D facial expression analysis was possible.

4.4.1 Performance

The recall rate, precision rate, and F$_1$-measure were calculated for each expression. This was first conducted using the automatically selected window widths, and then using the manually optimised window widths which show a significant increase in the average F$_1$-measure achieved. The full results achieved, including average, can be seen in Table 4.1. This table shows that the recognition rates for the different expressions varies widely. The best result achieved is for Surprise, with an F$_1$-measure of 82.56%, with the lowest rate found for Fear, with an F$_1$-measure of only 46.15%. The average rate found was 64.46%.

The confusion matrix produced for the six universal expressions using the manually selected window widths can be seen in Table 4.2. This shows the percentage of each expression that was correctly classified, along with where the misclassifications occurred. The confusion matrix shows where the main errors are introduced. Fear is often classified as Surprise, which is an expected result due to the similarities in the way these two expressions are acted - subjects often stretch their mouths, raise their eyebrows and open their eyes in both cases. However in addition, it is also regularly misclassified as Sad, which could be due to creasing around the eyes in both expressions which gives some similarities. The main confusion for Angry comes from incorrect classification as Sad, though this expression is often misclassified as Disgust as well. These anomalies could be due to the similarities in the creases in the forehead for these three expressions, specifically in
2D Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Happy</th>
<th>Sad</th>
<th>Angry</th>
<th>Disgust</th>
<th>Surprise</th>
<th>Fear</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>76.40</td>
<td>8.99</td>
<td>1.12</td>
<td>4.49</td>
<td>5.62</td>
<td>3.37</td>
</tr>
<tr>
<td>Sa</td>
<td>18.97</td>
<td>44.83</td>
<td>13.79</td>
<td>1.72</td>
<td>6.90</td>
<td>13.79</td>
</tr>
<tr>
<td>A</td>
<td>1.92</td>
<td>13.46</td>
<td>61.54</td>
<td>15.38</td>
<td>5.77</td>
<td>1.92</td>
</tr>
<tr>
<td>D</td>
<td>6.78</td>
<td>8.47</td>
<td>15.25</td>
<td>50.85</td>
<td>10.17</td>
<td>8.47</td>
</tr>
<tr>
<td>Su</td>
<td>0.00</td>
<td>4.65</td>
<td>3.49</td>
<td>0.00</td>
<td>89.53</td>
<td>2.33</td>
</tr>
<tr>
<td>F</td>
<td>2.56</td>
<td>20.51</td>
<td>10.26</td>
<td>2.56</td>
<td>25.64</td>
<td>38.46</td>
</tr>
</tbody>
</table>

3D Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Happy</th>
<th>Sad</th>
<th>Angry</th>
<th>Disgust</th>
<th>Surprise</th>
<th>Fear</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>75.28</td>
<td>2.25</td>
<td>1.12</td>
<td>6.74</td>
<td>4.49</td>
<td>10.11</td>
</tr>
<tr>
<td>Sa</td>
<td>1.72</td>
<td>68.97</td>
<td>17.24</td>
<td>5.17</td>
<td>3.45</td>
<td>3.45</td>
</tr>
<tr>
<td>A</td>
<td>0.00</td>
<td>28.85</td>
<td>51.92</td>
<td>11.54</td>
<td>1.92</td>
<td>5.77</td>
</tr>
<tr>
<td>D</td>
<td>3.39</td>
<td>5.08</td>
<td>16.95</td>
<td>62.71</td>
<td>5.08</td>
<td>6.78</td>
</tr>
<tr>
<td>Su</td>
<td>2.33</td>
<td>4.65</td>
<td>4.65</td>
<td>2.33</td>
<td>82.56</td>
<td>3.49</td>
</tr>
<tr>
<td>F</td>
<td>10.26</td>
<td>15.38</td>
<td>10.26</td>
<td>5.13</td>
<td>12.82</td>
<td>46.15</td>
</tr>
</tbody>
</table>

Table 4.2: Confusion matrices for 2D and 3D testing.

examples where the subject does not have much movement in other areas of the face such as the mouth. Disgust is most often misclassified as Angry, and to a lesser extent Sad, showing again the similarities in these expressions as seen by the 3D system. Happy is misclassified as Fear most often, which could be explained by the fact that the corners of the mouth move horizontally outwards in several examples of fear in this database.

The window widths employed for each expression are also shown in Table 4.1. These show that Happy, Sad and Angry perform best with a window width of 12, whereas the remaining three expressions give the best performance with a window width of only 8. In order to determine if these window widths are what would be expected, they are compared to the distributions of the length of the onset and offset segments for each expression, and the results shown in Fig. 4.8. These plots show that the window width chosen in each case generally falls within one standard deviation of the mean of either the onset or offset mean, and in almost all cases it lies within this range for both segments. Only for Disgust can the window width be observed to be below this range for the onset segment, and in this case it is only just within the range for the offset, though it does lie very close to the mode in this case. This plots show that the window widths chosen generally appear to be sensible compared to the onset, offset, or both, lengths.
Figure 4.8: Comparison between the chosen window widths and the onset and offset length distributions for each expression. Violin plots are exploited to show the number of sequences with each segment length, the mean is shown as a red cross at the centre of the red bar showing the standard deviation, and the window width chosen is shown as a yellow circled cross. (a) The onset length distributions for each expression (b) The offset length distributions for each expression.

4.4.2 Comparison to 2D

The $F_1$-measures achieved in the six-way decision when employing the 2D method, for both the automatic and manually selected window widths, can be seen in Table 4.1. The corresponding confusion matrix for the manually selected widths using the 2D method can be seen in Table 4.2. With both automatic and manual window width selection, the 3D system achieves a higher average $F_1$-measure than 2D: 61.03% compared to 59.70%, and 64.46% compared to 60.28%. In the manual case, for five of the expressions, Happy, Sad, Disgust, Surprise and Fear, the 3D method outperforms the 2D method, achieving a significant rise in $F_1$-measure. This is particularly striking for the Sad expression, which achieves a far higher $F_1$-measure with 3D features than with 2D, 62.5% compared to 44.83%. This improvement seen with the 3D data is in contrast to that seen purely from the validation results in Fig. 4.7. This suggests that the 3D system is not generally superior to 2D when distinguishing positive and negative examples for most of the expressions, with the notable exception of Sad, and to a lesser extent Fear. However, 3D information is beneficial when it comes to discrimination between the six expressions, which is demonstrated by the improvement in the $F_1$-measure seen in five out of six of
The only expression for which the 2D method significantly outperforms the 3D is Angry, achieving an $F_1$-measure of 58.72% compared to 50.00%. Comparing the performance in the validation results, the 2D system does demonstrate a slightly better ability to discriminate between positive Angry examples and the other negative examples. This may be because the FFDs employed in the 3D case are too coarse to pick up on the subtle motions in the forehead and around the eyes that are present in many of the Angry examples. However, the significant difference in the Angry results in the six-way decision are not accounted for solely by this. In addition, most of the misclassification of this expression seems to be due to the performance of Sad. Though the 2D system misclassifies a significant number of Angry sequences as Disgust, as happens in the 3D case, the difference here is that there is much less confusion between Angry and Sad. The reason for this may be due to the fact that in this case Sad performs very poorly for 2D. This suggests that the 3D process is able to distinguish features that are useful for recognition of the Sad expression that are not possible using 2D. This increases the recognition rate for Sad, but it could mean that these features are confused with those present in the Angry sequences, and so add to the confusion between these two expressions which results in a much lower classification rate for Angry.

4.4.3 Temporal Comparison

The proposed method employs HMMs in order to temporally model the full expression, and to find the most likely sequence from the individual frame outputs from the GB classifiers. The aim of this process is to smooth errors in the GB classification when predicting the frame sequence, to ensure that the full sequence is present in order for any part of the sequence to be labelled, and then to require that the sequence is likely enough to be chosen from the classifier outputs. In addition, the likelihood of each of the sequences being predicted is also exploited to determine the best expression for each sequence.

The smoothing benefits of this method are demonstrated in Fig. 4.9. This figure shows positive and negative examples from the Sad classifier results. The graphs in 4.9a and 4.9b
4. Dynamic 3D Facial Expression Analysis

Figure 4.9: The Sad GentleBoost classifier outputs and labels for positive (true Sad sequence) and negative (Fear sequence) examples. Frame labels: black - neutral, blue - onset, yellow - apex, red - offset. (a) Sad onset and offset classifier outputs for a Sad sequence (b) Sad onset and offset classifier outputs for a Fear sequence (c) True Sad classifier segment labels for Sad sequence (d) True Sad classifier segment labels for Fear sequence (e) Sad GB classifiers predicted labels for Sad sequence (f) Sad GB classifiers predicted labels for Fear sequence (g) Sad HMM classifier predicted labels for Sad sequence (h) Sad HMM classifier predicted labels for Fear sequence
show the emission values from the two GB classifiers for a positive example, Sad, and a negative example, Fear, respectively. As has been previously stated, these emission values are formed from the labels for the frames, 1 or \(-1\), being multiplied by the confidence value for these labels. Hence, whenever the plot becomes positive, this is due to the label changing from negative to positive classification. Alongside these, Figs. 4.9c and 4.9d show the true frame segment labelling for these sequences for the Sad classifier, Figs. 4.9e and 4.9f show the frame labels taken directly from each of the classifiers, with the apex frames inferred as filling in gaps between the onset and offset frames, and Figs. 4.9g and 4.9h show the labels predicted by the most likely HMM sequence when using the emission values from the classifiers.

The positive example demonstrates that it can be possible for the GB classifiers to repeatedly classify onset and offset frames throughout the sequence, as is shown in Fig. 4.9e, but the HMM takes only the main sections of the sequence where these classifications occur (where the emission values are at their highest) as part of the most likely sequence, Fig. 4.9g. Hence the effect is to smooth the labelling to something that is much closer to the true frame labels as shown in Fig. 4.9c. The negative example demonstrates the other benefit of the HMM. This time the GB classifiers classify some frames as onset and offset, as seen in Fig. 4.9f, even though none are present for the Sad expression in this sequence. This results in apex frames being inferred, so this expression would be considered to possibly be Sad if using the classifiers outputs alone. However, the HMM is able to smooth over these frames, choosing the most likely sequence as one that contains only neutral frames, as seen in Fig. 4.9h, due to the low emission values, and small number, of these frames. This means that this sequence would be rejected immediately as not being Sad.

In addition to looking at particular examples of the benefit of using the HMMs for temporal modelling, it is also possible to do an analysis of the expression classification differences between using the GB classifier outputs directly, from the HMM classifier outputs. Due to the way the six-way decision is made from the sequence likelihoods, and the fact there is no equivalent probability measure to use for the GB classifiers, it is not possible to perform a fair comparison between six-way expression classifications directly. However,
4. Dynamic 3D Facial Expression Analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>Pos</th>
<th>Neg</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBs</td>
<td>92.4</td>
<td>36.0</td>
<td>64.2</td>
</tr>
<tr>
<td>GBs + HMM</td>
<td>80.4</td>
<td>72.6</td>
<td>76.5</td>
</tr>
</tbody>
</table>

Table 4.3: Percentage of positive/negative sequences containing/not containing the apex state using the GB classifier outputs directly compared to the HMM outputs for a window width of 12.

One way to compare the performance of each method is to assess the percentage of sequences that are predicted to contain the apex state in each case. This would be the first stage in each classification process, where those expressions for which the sequence does not display the apex state are rejected. Here the desired result is examined for positive and negative examples for each expression: positive examples should contain the apex, and negative examples should not contain the apex state. The results of this analysis for a window width of 12 can be seen in Table 4.3. These results show that the GB classifiers generally show the apex for a higher percentage of positive examples than the HMM classifier - 92.4% compared to 80.4%. However, this is at a cost, as the negative rates show. The GB classifiers also give the apex state in a large number of negative sequences, with only 36.0% of sequences not containing the apex state. This is compared to 72.6% when using the smoothed HMM output. This results in the HMM process giving a much higher average percentage of 76.5% compared to the average GB result of 64.2%. This demonstrates that the full HMM classifier allows better discrimination between positive and negative examples purely on the basis of whether the apex is present or not, and that is before the likelihoods of the different expressions are taken into account.

4.4.4 Summary and Discussion

This method has been demonstrated to exploit the extra information available in the 3D facial geometries to improve on the results found with the 2D, and to employ temporal modelling to smooth over incorrect classifications from the GB classifiers in order to correctly classify image sequences. The approach proposed in this chapter has been shown to achieve an improvement over the equivalent 2D method when tested on the BU-4DFE database. The utilisation of HMMs was shown in 4.4.3 to be beneficial in capturing more accurate temporal aspects of the facial expressions, as well as reducing the number of
false positives over those captured by the GB classifiers alone.

The results show that there are a number of image sequences will are still incorrectly classified by this method. The main way in which the method fails is when an expression is classified as positive by two or more expressions. This can occur due to the onset/offset being incorrectly detected by some of the GB classifiers, and the expression which the highest likelihood is then one of the incorrect expressions. This may happen due to variability in the motion during the onset/offsets of some of the expressions, and the GB classifiers not capturing this variability adequately.

Another failure mode is when the onset or offset for an expression is not detected by the appropriate classifier, and so another expression is deemed most likely. These problems occur both in the 2D and 3D methods, and demonstrate the the GB classifiers are not capable of fully capturing the variability of the onset and offsets for all the expressions. Throughout the database, the same expression can be acted in very different ways by different subjects, hence giving a wide range of features that are particular to the onset or offset of an expression, and this causes these problems. Even within the onset of an expression by one subject, there can be different stage (e.g. the upper face and lower face moving separately). In order to deal with this problem a more complex model would be required that is able to capture the different aspects of the onset or offset of an expression. However, the complexity required would quickly make this method unwieldy, especially as more expressions were added to the recognition process. For this reason in future work should focus towards AU detection rather than full expression analysis.

In addition, is it possible that the low recognition rate of some of the expressions was due to the fact that the features employed here might not have been optimal for capturing the information required for classification of onset/offset of these expressions. An exploration of other motion-based features, such as higher-order measures, could therefore be undertaken to identify if there is a better way to capture the facial motion needed for this purpose. Alternatively, the use of different features altogether, such as dynamic variants of the binary pattern based features employed in Chapter 3 might be expected improve performance, and thus should be explored in future work.
4. Dynamic 3D Facial Expression Analysis

There are other extensions to this method that would be desirable in future work. Posed expressions, as employed in these experiments, have been shown to differ greatly, in content and dynamics, from spontaneous natural expression data. Spontaneous expressions are also rarely seen on their own, subjects often display a mixture of emotions, such as amusement and embarrassment, or sadness and anger. In addition, one expression can transition to another without a neutral expression in between, and expressions can be mixed with periods of speech. In order to create a system that will be useful in real-life situations, it is therefore highly desirable to train and test this system on this kind of data, and to adapt the models employed in order to do so.

4.5 Conclusions

In this chapter, the facial geometry data in the BU-4DFE database was exploited to perform dynamic analysis of the six universal expressions for the purpose of fully automatic expression recognition. The methodology employed 3D motion-based features, captured with FFDs, which were captured in each pair of dimensions, spatial and time. Best features were chosen and classified by GB classifiers, and the output of these was exploited to build temporal models of each expression using an HMM. Six-way classification was conducted using all sequences in the database deemed to accurately reflect the expression, with a classifier being trained and tested for each expression in each fold. Window widths were chosen based on the validation results for each expression.

The results were compared with the same method (using manual alignment) conducted on 2D facial motion data extracted from the facial intensity image sequences in the the same database. The expression recognition rates achieved indicate that there is a gain when using 3D facial geometry data, and that the 3D data is particularly important for correct classification of the Sad expression. In addition, temporal analysis indicates that modelling the full dynamics of the expression with the HMMs leads to a higher recognition rate of the expressions than using the GB classifier outputs alone.
Facial Action Unit Intensity Estimation

Preliminary results for this work were published as [SZP13].

5.1 Introduction

Detection of AUs alone is an important source of information about the full expression, and thus emotional state, of a subject. However, knowing the full intensity of the AUs in an image or video greatly increases the richness of the information, allowing more complex emotional states to be inferred. For example, in the application of pain detection, the intensity level of a subset of AUs has been shown to be important in determining the complex states, such as the level of pain in a subject [125].

Accurate AU intensity estimation is a difficult problem, particularly in the situation where more than one AU is present, which can considerably alter the appearance of each individual AU. But, usefully, AUs are known to display a large number of patterns and correlations, and certain combinations are very regularly seen in typical expression behaviour. For example, the combination of AU 1 and AU 4 is very common in worried expressions. Examples of each of these AUs alone and together can be seen in Fig. 5.1. As can be seen from this example, AU 1 appears very differently in Fig. 5.1a to how it looks in Fig. 5.1c when combined with AU4. Though there are still traces of the distinctive forehead furrows, these are largely masked by the deep creases between the eyebrows, making the AU more difficult to detect. But the common occurrence of this combination
can be exploited to overcome this problem.

In this chapter, first techniques for estimation of the intensities of AUs are explored. In particular, two regression techniques are compared, SVRs that have previously been applied to this problem, and a new approach - the application of multiclass Direct Acyclic Graph SVMs (DAG-SVMs) in place of regressors, where each intensity of the AU is taken as a separate class, and the classifiers trained appropriately. The performance of two different feature selection methods are also investigated on this problem.

In order to exploit the correlations between AUs to improve on individual estimations, a MRF tree switching method is then investigated. This approach aims to model the correlation links between AU intensities in a face region in a simple way that allows for simple, exact inference. Multiple trees are employed, each modelling the case where one AU has maximum intensity, to allow for the correlations in different expression contexts. This approach allows choosing the most likely tree, and hence deducing the expression context whilst simultaneously estimating the full set of AU intensities.

The contributions of this chapter are as follows:

- The application of DAG-SVM regressors for AU intensity estimations.
- Application of MRF tree structures to harness AU correlations for improved intensity estimation.
- The exploitation of multiple trees to allow for optimal structure selection based on context.

This chapter is organised as follows: Section 5.2 details the two regression techniques, and training methodology, including feature selection alternatives. Section 5.3 then outlines the model framework for exploitation of AU structural information. In Section 5.4 the experiments conducted to explore the relative performance of regression and model are described, and finally 5.5 concludes the chapter.
5.2 Regression for Intensity Estimation

In this section, the comparative performance of two different regression techniques are explored. The feature extraction and selection methods are described, along with two methods for choosing regions in the face image from which to choose the features.

5.2.1 Feature Descriptor

The first stage in the system is to use the given facial landmarks to perform alignment of the images. Exploiting the calculated positions of the eyes and nose, the images are transformed into a predefined frame to ensure alignment of facial points suitable for feature extraction.

2DLGBPs features [214] are employed as the feature type, as described in Section 2.3.1. Both the magnitude and phase components of the feature are utilised, with the image region divided into 10x10 blocks in order to construct the feature descriptor via concatenation of the block histograms.
5. Facial Action Unit Intensity Estimation

5.2.2 Feature Selection

Feature selection is then performed in order to extract the most discriminative subset of features. The GB algorithm is employed for this purpose. Before applying individual feature selection, a number of blocks are preselected in order to focus on the areas in the image of most interest for each AU from which to choose these features. Two methods are utilised for this purpose:

- **Predefined Region**: Take a number of regions that would be expected to contain the relevant information about the AU. As AUs are naturally divided into upper and lower face AUs, the image is divided into upper and lower face, and only the appropriate features are taken.

- **Selected Region**: Allow the most discriminative regions to be chosen automatically. The GB method for region selection is employed to choose the most discriminative regions from the entire image for each AU.

The region selection algorithm, Algorithm 2, is based on the individual feature selection GB algorithm, and was previously described in Section 3.5.2. The full algorithm description is included here for completeness. Each region is taken in turn and the individual features extracted from this region are used to calculate a least-squares regression on the training data and labels. The error is summed for all regions, and the one that produces the lowest error rate is chosen as the most discriminative in this pass. Then the predicted labels when using the features in this region are averaged to give an estimated label, and this is used to update the weights for all examples in the training set, allowing examples that are misclassified to be focussed on (boosted) in the next pass as they become more highly weighted.

Once the region has been selected, traditional GB feature selection is then applied to choose individual features. As before, avoid over-fitting, the strategy is to run the selection algorithm repeatedly, removing the previously chosen features at each stage, until the number of features selected exceeds the chosen threshold, set to 200 in this case. The input to the feature selection algorithm is taken as the AU classification labels, i.e. presence or
5.2. Regression for Intensity Estimation

Figure 5.2: The DAG-SVM Operation. Each node consists of one binary SVM classifier that divides the feature space as shown. As testing examples are filtered through the graph, the possible classes are decreased until a final classification is made.

absence of the AU, rather than intensity values. This was shown to give a better set of discriminative features for use in regression.

5.2.3 Regression

Two forms of regression are investigated here: Firstly, the standard SVR technique, and secondly, multiclass DAG-SVMs applied with each intensity trained as a separate class. These are approaches are each described in turn. Both regressors are trained per AU. They are first parameter optimised using three-fold cross-validation on a portion of a validation set, which is chosen to be AU specific. Then they are trained on the remaining portion of the validation set intensity values for the AU, including labels of zero to represent absence.
5. Facial Action Unit Intensity Estimation

**Support Vector Regression** SVRs aim to fit a central prediction function to the training data such that the margin between this function and support vectors in the training examples is minimised.

Let \( \{x_1, ..., x_M\} \) be the input training data for \( M \) frames, with corresponding intensity labels, \( \{y_1, ..., y_M\} \), for one AU. The simple form of the SVR problem constructs a linear function such that:

\[
f(x) = w^T x + b
\]  

(5.1)

where \( w \) is a coefficient vector, and \( b \) is a threshold.

The function is then fitted through solving the following optimisation problem:

\[
\begin{align*}
\text{minimise} & \quad \frac{1}{2}||w||^2 + C \sum_{i=1}^{M} (\xi_i + \xi_{M+i}) \\
\text{subject to} & \quad y_i - f(x_i) \leq \epsilon + \xi_i, \\
& \quad f(x_i) - y_i \leq \epsilon + \xi_{M+i}, \\
& \quad \xi_i, \xi_{M+i} \geq 0
\end{align*}
\]  

(5.2)

where \( \epsilon \) is the allowed error, and \( \xi_i \) and \( \xi_{M+i} \) are slack variables for data \( x_i \) that exceeds this error. The constant \( C \) controls the smoothness of function \( f \). In practise, the feature vector can be transformed into a high dimensional space through the use of nonlinear functions. And in the the dual formulation this can be done through employing kernel functions to measure similarity between the training examples. Here as before, due to the form of the features chosen, the histogram intersection function is employed for this purpose, which takes the form of \( k(x_i, x_j) = \sum_{n=1}^{N} \min(x_{in}, x_{jn}) \)

**Directed Acyclic Graph Support Vector Machines** DAG-SVMs use a number of binary 1vs1 SVM classifiers each of which acts as a node in the graph. They act as hyperplanes, dividing the test examples into gradually finer classes, until a single classification is selected [124]. At each node, a binary SVM is employed, trained on two classes (e.g. 0 vs 5 at the first node). Because the data at this stage contains not only examples of 0 and 5, but all classes in between, it is only possible to say that if it is not classified as 0, then it is 'Not 0', and vice-versa for 5. Then at the next node data that was classified as 'Not 0' is further narrowed down to be either 'Not 5' or 'Not 1'. In this
way as the examples pass through the graph their classification is gradually restricted until they can finally be labelled in the final row. Fig. 5.2 demonstrates this operation.

Each binary classifier is trained according to the traditional SVM operation, as described in Section 3.5.3. Here, the training data is divided appropriately to train each SVM on two of the intensity classes only, including no presence as an additional class. Testing then proceeds via an efficient algorithm that tracks the route of each example through the graph of classifiers. This can be seen in Algorithm 5. A route matrix, $R$, is employed to track the paths. The columns of this matrix represent each example, and rows whether that example has been diverted away at each stage from taking the left hand path (as shown in Fig. 5.2) from the current node. So for example, an example of true intensity 1 would not be diverted from this path until the penultimate node (0 vs 2), and so the column of $R$ for this example would be $(0, 0, 0, 0, 1)^T$, if the example was correctly classified. Identifying the final classification is then trivial - it can be done by summing the columns of the matrix $R$. Each of the binary classifiers also employs a histogram intersection kernel as in the SVR case.

### 5.3 An Intensity Markov Random Field Structure

In order to enhance the results from regression alone, information about the correlations between AUs can be harnessed. Here a novel approach based on a set of MRF trees is proposed, for modelling combinations of AU intensities within videos of a particular face region. This approach is based on two assumptions about facial expression behaviour: first, that it is expected that the highest intensity AU in a face region will impact on the intensities present of the other AUs, and signify a particular expression context, and second, that it is expected there will be correlations between the AUs in this context.

To exploit these assumptions, each frame in the video is modelled as a MRF tree, with the maximum intensity AU in that frame as the root of the tree. Given the choice of trees, the aim is to predict which tree is most likely for that frame (i.e. which AU has highest intensity) and simultaneously infer the most likely intensity combination is of all AUs in the face region.
Algorithm 5 The DAG-SVM testing operation

Require: Testing set \( X \) with true labels \( y \)
Require: Number of intensity states \( N \)
Require: Graph containing \( L \) levels - a particular level, \( l \), will contain \( l \) nodes

Initialise route matrix \( R \) to track paths through graph, set to be zero for all links

for each depth level \( l \) in graph do
    for each node \( p \) at level \( l \) do
        Calculate class indices at node:
        \[
        c_1 = p, c_2 = N - l + p
        \]
        Find examples, \( X_p \), with routes that lead to this node which consists of feature vectors \( x_k \) for which
        \[
        \sum_{i=1}^{l-1} r_{k,i} = p
        \]
        Test \( X_p \) against binary classifier for classes \( c_1 \) and \( c_2 \) to produce labels \( \hat{y}_p \)
        For each label where \( \hat{y}_q = -1, \hat{y}_q \in \hat{y}_p \) set appropriate route matrix element \( r_{q,l} = 1 \)
    end for
end for

Sum elements of \( R \) to identify final classification label for example \( x_k \).
\[
\hat{y}_k = \sum_{i=1}^{L} r_{k,i}
\]

In this section the face region tree model of AU intensities is defined. Let \( I \) be the relevant face region, with feature descriptor \( x \), which contains \( N \) AU parts with intensities \( \Lambda = \{\lambda_1, ..., \lambda_N\} \). A set of part-based models, \( M = \{T_1, ..., T_N\} \), are built, each of which takes the form of a tree, \( T_t = (V, E_t) \). In these graphs, the vertices \( V = \{v_1, ..., v_N\} \), are the AU parts that take an intensity value (which is set to zero if absent) within the image region with an intensity ranging from A-E. The total set of parts is common to all region trees, but in each case \( v_t \) is set to be the root AU for tree \( T_t \). Each tree has a set of edges \((v_i, v_j) \in E_t\), which connect pairs of these parts. These can be thought of as springs which are stretched by varying degrees depending on the difference in intensities between the two AU parts, \( v_i \) and \( v_j \).
5.3. An Intensity Markov Random Field Structure

Facial Region Video Sequence

Most Likely Maximum AU Intensities

Most Likely AU Intensity Combination

Figure 5.3: A set of Markov Random Field trees are built to model the AU intensity combinations in a face region, and the predicted intensities calculated for each tree. The most likely tree is then chosen at each frame.

5.3.1 Face Region Model

The probabilistic model allows the assumption as random variables of the intensities of a number of AUs in a certain facial region, Λ in a particular frame. Assume the results of applying a set of functions, \( F = \{f_1, ..., f_N\} \), are applied to the feature descriptor, \( x \), of the image region are known. The aim is to find the most likely tree with particular intensity combination.

The joint posterior probability of tree index \( t \) having the combination of intensities \( \Lambda \) can then be formulated as:

\[
p(\Lambda, t|F(x)) = p(\Lambda|F(x), t)p(t|F(x))
\]  

(5.3)

where \( p(\Lambda|F(x), t) \) is the likelihood of the intensity combination of tree \( t \) given \( F(x) \), and \( p(t|F(x)) \) is the conditional probability of tree \( t \) given \( F(x) \). From here on \( F(x) \) and \( f_i(x) \) will be abbreviated as \( F \) and \( f_i \) for brevity purposes.
This can then be further written as:

\[ p(\Lambda|F, t) = \frac{p(F|\Lambda, t)p(\Lambda|t)}{p(F|t)} \quad (5.4) \]

where \( p(F|\Lambda, t) \) is the likelihood of the feature descriptor given the intensity configuration \( \Lambda \) for tree \( t \), and \( p(\Lambda|t) \) is the intensity combination joint prior distribution over AU intensities for tree \( t \). Hence, as \( p(F|t) = p(t|F)p(F)/p(t) \), the posterior can be simplified to be:

\[ p(\Lambda, t|F) = p(F|\Lambda, t)p(\Lambda|t)p(t) \quad (5.5) \]

The likelihood probability for \( F \) given the intensity values of the individual parts of the tree and the corresponding parameters of the model, can then be defined. It is assumed that the likelihood of each part intensity, \( \lambda_i \), will be dependent on the value of the function \( f_i \) for that part. Hence, this gives:

\[ p(F|\Lambda, t) = \prod_i p(f_i|\lambda_i, t) \quad (5.6) \]

where \( p(f_i|\lambda_i, t) \) is the likelihood of the individual feature descriptor given that AU part \( v_i \) has an intensity \( \lambda_i \). The particular choice of likelihood functions is described in the next section.

The prior probability, \( p(\Lambda|t) \), models the relationships between AU intensities for a particular tree, and can also be simply split, as in the general form for a MRF:

\[ p(\Lambda|t) = p(\lambda_t|t) \prod_{(v_i, v_j) \in E_t} p(\lambda_j|\lambda_i, t) \quad (5.7) \]

where \( p(\lambda_t|t) \) is the prior probability of the root of the tree, \( t \), and \( p(\lambda_j|\lambda_i, t) \) is the conditional probability of the child intensity \( \lambda_j \) given parent intensity \( \lambda_i \), in this tree. Thus the posterior distribution becomes:

\[ p(\Lambda, t|F) = p(t)p(\lambda_t|t) \prod_i p(f_i|\lambda_i, t) \prod_{(v_i, v_j) \in E_t} p(\lambda_j|\lambda_i, t) \quad (5.8) \]

By taking negative logarithms of both sides this is used to define an energy minimisation function for each model that must be minimised in order to match the model to an image.
5.3. An Intensity Markov Random Field Structure

region. The following components are set:

\[ s^t(\lambda_t) = -\log(p(t)p(\lambda_t|t)) = -\log(p(\lambda_t, t)) \]

\[ a^t_i(\lambda_i) = -\log(p(f_i|\lambda_i, t)) \]

\[ c^t_{ij}(\lambda_i, \lambda_j) = -\log(p(\lambda_j|\lambda_i, t)) \]

(5.9)

Therefore, equation 5.8 can be written as:

\[ G^t(\Lambda) = s^t + \sum_i a^t_i(\lambda_i) + \sum_{(v_i,v_j) \in E_t} c^t_{ij}(\lambda_i, \lambda_j) \]

(5.10)

This equation gives the energy function that when minimised will give the most likely intensity combination for this tree and overall probability of tree and combination. It consists of three components: \( s^t \) which is defined as a selection function for tree \( t \), \( a^t(\lambda_t) \) is defined as an appearance function that measures mismatch for an image region when each part \( v_i \) is given an intensity of \( \lambda_i \), and \( c^t_{ij}(\lambda_i, \lambda_j) \) is defined as a combination function that measures the mismatch in the deformation of the model given the image region, when connected parts \( v_i \) and \( v_j \) have intensities of \( \lambda_i \) and \( \lambda_j \). The optimal intensity values are thus given by minimising this function.

5.3.2 Tree Structures

In order to create meaningful structures that are able to model the interdependancies between AUs, without adding loops, a set of tree graphs are employed, \( \{T_1, ..., T_N\} \), each of which has the equivalent AU part, \( \{v_1, ..., v_N\} \), as the root node, as shown in Fig. 5.3.

The aim of this is to allow each tree to best model the cases where the root AU has the highest intensity, as this is when it is expected that it will most impact on the appearance of the other AUs in the region. For this reason, only training examples where this is the case are utilised to construct each tree. There are many ways to learn the tree structures, here an adapted version of the Chow-Liu algorithm [27] is exploited to build the tree structures.

First the training labels are employed to calculate the mutual entropy between all pairs of AUs. Two sets of parts are defined: \( V_T \) are the parts in the tree, and \( V_L \) are the parts
still to add, where $V_T \cup V_L = V$. Starting from the root $AU$, which is $v_t$ for $T_t$, this is added to $V_T$, and $V_L$ is set to consist of all other parts. In the first step, the two parts in $V_L$ which have the two highest mutual entropy scores with the root are taken. They are added as child nodes of $v_t$, and are moved from $V_L$ to $V_T$. Edges connecting these parts are also added to $E_t$. This step ensures that the root has at least two children, which improves the impact of the other parts on the root node. Then the highest mutual entropy score, for any pair of nodes such that $v_p \in V_T$ and $v_c \in V_L$, is identified. $v_c$ is then added as a child of $v_p$ in the tree, the edge $v_p, v_c$ added to $E_t$, $v_c$ removed from $V_L$ and added to $V_T$. The algorithm then repeats this step until all parts have been added into the tree, and $V_L$ is empty.

5.3.3 Model Parameters

Starting with the tree structure for tree $t$, the training set $L^t$, which contains all examples for which $v_t$ has the highest intensity, is utilised to calculate the parameters of the tree. This consists of computing each component of the energy function in Equation 5.10, for all $F$s. These are all calculated in similar ways, using the training data directly to form conditional probability distributions. This method assumes discrete intensity labels. However, it could be extended to deal with continuous intensity labels by taking ranges of values around each integer label.

Selection Function The selection function, $s^t(\lambda_t) = -\log(p(\lambda_t, t))$ is simply the joint prior probability of tree $t$ with root part $v_t$ having intensity $\lambda_t$. This can be calculated directly as:

$$s^t(\lambda_t) = \left( \frac{|L^{t \leftarrow \lambda_t}|}{|L|} \right)$$

(5.11)

where $|L^{t \leftarrow \lambda_t}|$ are the number of tree $t$ training examples for which the root part has intensity $\lambda_t$, and $|L|$ is the total number of training examples.

Appearance Function It is necessary to define $a^t_i(\lambda_i) = -\log(p(f_i|\lambda_i, t))$, for each part $v_i$, over the the range of function outputs. This is modelled as a Gaussian distribution, with mean $\mu^t_{i,\lambda_i}$ and standard deviation $\sigma^t_{i,\lambda_i}$. The distribution is calculated for each
5.3. An Intensity Markov Random Field Structure

possible intensity value that the part can take:

\[ a_t^i(\lambda_i) = -\log \left( \frac{1}{\sqrt{2\pi\sigma_{t,i,\lambda_i}^2}} \exp \left( \frac{-(f_i - \mu_{t,i,\lambda_i})^2}{2\sigma_{t,i,\lambda_i}^2} \right) \right) \]
\[ = \frac{1}{2} \log(2\pi) - \frac{1}{2\sigma_{t,i,\lambda_i}^2} (f_i - \mu_{t,i,\lambda_i})^2 \]  

(5.12)

Hence, the parameters of this distribution for each possible intensity value, \( \lambda_i = \{0, ..., 5\} \):
\[ \theta_i^t = \{\mu_{t,i,0}, \sigma_{t,i,0}^2, ..., \mu_{t,i,5}, \sigma_{t,i,5}^2\} \], are calculated to allow calculation of the appearance likelihoods.

In order to calculate distributions for the cases for which there are no examples in the data, the distribution means are interpolated to fill in missing values, and missing deviation values are taken as the mean of those calculated from the data.

**Combination Function**  It is necessary to define \( c_{i,j}^t(\lambda_i, \lambda_j) = -\log(p(\lambda_j|\lambda_i, t)) \) for each connected pair of AU parts. For this purpose, the conditional probability of each child intensity for each possible intensity value of the parent part, is determined, given the data. So for each edge \((v_i, v_j) \in E_t\), this can be calculated as:

\[ c_{i,j}^t(\lambda_i, \lambda_j) = -\log \left( \frac{|L_{v_j=\lambda_j} \cap L_{v_i=\lambda_i}^t|}{|L_{v_i=\lambda_i}^t|} \right) \]  

(5.13)

Laplacian smoothing is applied to this distribution to account for missing data.

**5.3.4 Intensity Inference**

In order to minimise Equation 5.10, a method based on the well known Viterbi algorithm, and exploited in [49] for efficient inference on MRF structures, is employed.

Starting from a leaf node in the tree (i.e. a part with no children), \( v_j \), the best intensity value, \( \lambda_j \), given each possible intensity value of its parent part, \( \lambda_i \), can be computed by minimizing the appearance and combination mismatches at each parent intensity:

\[ m_j^t(\lambda_i) = \min_{\lambda_j} (c_{i,j}^t(\lambda_i, \lambda_j)) \]  

(5.14)

and storing the intensity value \( \lambda_j \) at each parent intensity.
This minimum energy can then be taken as message from the child node, which summed
together across all children contribute to the parent energy function. If \( v_j \) is now a non-
root part with children, \( C_j \), then the function to minimise becomes:

\[
m^t_j(\lambda_i) = \min_{\lambda_j} \left( c^t_{i,j}(\lambda_i, \lambda_j) + \sum_{v_c \in C_j} m^t_c(\lambda_j) \right)
\]  

(5.15)

and the intensity values \( \lambda_j \) can again be stored for each parent intensity.

Finally the optimal root intensity can be computed by summing all messages from child
parts with the appearance score and minimising across possible intensity values:

\[
G^t(\lambda_c) = s^t + \min_{\lambda_i} \left( a^t(\lambda_i) + \sum_{v_c \in C_i} m^t_c(\lambda_i) \right)
\]

(5.16)

The optimal intensities can then be found for all parts in the tree by backtracking back
down the tree using the parent intensity to identify the best intensity value of each part.
5.3. An Intensity Markov Random Field Structure

5.3.5 Implementation

An overview of the implementation of the system can be seen in Fig. 5.4. This shows the full training process: feature extraction and selection, tree construction, model parameter calculation. Testing is then shown: message passing within all trees, maximum likelihood tree selection for each frame in the sequence, and finally taking the AU intensities from the chosen trees.

The training subjects are divided into validation and training sets. The former is employed to form AU specific balanced (equal number of positive (intensity > 0) and negative (intensity = 0) frames are taken) data for feature selection and parameter optimisation, and the latter for training and testing the regressors. Only half of the training set is employed for training, where a balanced set of frames are again taken to train the regressor. The full set of training subjects is then taken for testing, and in this case all frames are included. Including both seen and unseen data for testing builds a range of information into the model probability distributions about the regressor behaviour, so that it can interpret the (unseen) testing data. The regressor outputs from testing are then employed to calculate the parameters for each tree.

One regressor is trained for each part. They are first parameter optimised using threefold cross-validation on the validation set. In the case of the SVRs, the output from the regressors is then normalised in order to make it suitable for use as input to the MRF model. First, to remove a bias added by the regressors, the subject-specific mean is subtracted. Then scaling is applied according to the ratio of the standard deviation of the regressor versus that of the training labels. The testing output also has the same processing applied (subtraction of the mean and then the same scaling factor applied).

Let \( r_{i,n} \) be the output of one regressor trained for part \( v_i \), on subjects with training labels \( L_i \), and then tested on only subject \( n \). A subject specific function \( f_{i,n} \) is then defined such that:

\[
 f_{i,n} = k_i (r_{i,n} - \mu_{r_{i,n}}) \tag{5.17}
\]

where

\[
 k_i = \sigma_{r_i} / \sigma_{L_i} \tag{5.18}
\]
5. Facial Action Unit Intensity Estimation

<table>
<thead>
<tr>
<th>Region</th>
<th>AU</th>
<th>PCC</th>
<th>ICC</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SVR</td>
<td>DAG-SVM</td>
<td>SVR</td>
</tr>
<tr>
<td>R1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>0.51</td>
<td>0.50</td>
<td>0.45</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.49</td>
<td>0.51</td>
<td>0.40</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.55</td>
<td>0.66</td>
<td>0.48</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.39</td>
<td>0.59</td>
<td>0.33</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.51</td>
<td>0.52</td>
<td>0.43</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>0.20</td>
<td>0.21</td>
<td>0.16</td>
</tr>
<tr>
<td>R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>0.76</td>
<td>0.72</td>
<td>0.66</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>0.10</td>
<td>0.17</td>
<td>0.09</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>0.26</td>
<td>0.15</td>
<td>0.21</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.07</td>
<td>0.04</td>
<td>0.05</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>0.72</td>
<td>0.79</td>
<td>0.56</td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>0.37</td>
<td>0.38</td>
<td>0.32</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td>0.41</td>
<td>0.44</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Table 5.1: Regression results with predefined regions.

<table>
<thead>
<tr>
<th>Region</th>
<th>AU</th>
<th>PCC</th>
<th>ICC</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SVR</td>
<td>DAG-SVM</td>
<td>SVR</td>
</tr>
<tr>
<td>R3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>0.53</td>
<td>0.57</td>
<td>0.50</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.56</td>
<td>0.52</td>
<td>0.47</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.53</td>
<td>0.63</td>
<td>0.42</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.53</td>
<td>0.43</td>
<td>0.51</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.52</td>
<td>0.46</td>
<td>0.45</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>0.27</td>
<td>0.22</td>
<td>0.22</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>0.74</td>
<td>0.72</td>
<td>0.63</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>0.17</td>
<td>0.19</td>
<td>0.16</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>0.24</td>
<td>0.14</td>
<td>0.20</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.10</td>
<td>0.03</td>
<td>0.08</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>0.72</td>
<td>0.75</td>
<td>0.57</td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>0.44</td>
<td>0.39</td>
<td>0.38</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td>0.45</td>
<td>0.42</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Table 5.2: Regression results with selected regions.

given $\mu_{r,n}$ is the mean of the regressor only over frames for subject $n$, $\sigma_{r_i}$ is the standard deviations of the regressor output for part $v_i$ over the full training set, and $\sigma_{L_i}$ is the standard deviation of the training labels.

Tree structures are then constructed, as described in Section 5.3.2, and finally, the model parameters are then calculated as described in 5.3.3, using the tree structures that have been constructed
5.4 Experiments

Experiments were conducted on the Denver Intensity of Spontaneous Facial Actions (DISFA) database [99], one of only two naturalistic databases that have been FACS coded with AU intensity values. The only other similar database available is the UNBC-MacMaster Shoulder Pain Expression Archive Database [94], however this contains codings for only 9 AUs, and consists of a smaller variation of expressions due to its focus on pain. The intensities are recorded as values between 0-5, where 0 denotes the absence of the AU, and 1-5 represent A-E intensities. This database consists of 27 subjects, each recorded whilst watching a 4-minute (242 seconds) video clip by two cameras, left and right. The FACS coding included consists of 12 AUs: 1, 2, 4, 5, 6, 9, 12, 15, 17, 20, 25, 26. 66 facial landmarks are provided for each frame in each video. In this work only the left camera view was utilised, and the available landmarks exploited in order to align all frames by transforming images in order to match the eye and nose locations to an ideal position. There are a number of frames in each video for which the automatic landmarking method was known to have failed, thus resulting in the landmarks provided for these frames being inadequate for alignment. In the experiments these frames were simply removed from all data sets.

Finally, the predicted labels were smoothed using a moving average technique for all experiments reported here. This was shown to improve results in cases, by employing temporal information in the simplest way possible, and exploiting the fact that the intensities in consecutive frames for a particular AU will be highly correlated.

5.4.1 Regression

The performance of the two regression methods - SVRs and DAG-SVMs - was compared. These experiments were conducted using the two different regions outlined in Section 5.2.2: predefined regions, and selected regions. For the former, the image is split into two face regions, upper and lower face, and trained regressors for the appropriate AUs on each region: Region 1 (Upper Face): 1, 2, 4, 5, 6 and 9, and Region 2 (Lower Face): 12, 15, 17, 20, 25, 26. Alternatively, selected regions are chosen from the entire images, and then individual features selected from these, in order to train regressors for all AUs. The
5. Facial Action Unit Intensity Estimation

<table>
<thead>
<tr>
<th>Region</th>
<th>AU</th>
<th>PCC</th>
<th>ICC</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DAG-SVM</td>
<td>MRF</td>
<td>DAG-SVM</td>
</tr>
<tr>
<td>R1</td>
<td>1</td>
<td>0.50</td>
<td>0.51</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.51</td>
<td>0.43</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.66</td>
<td>0.63</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.59</td>
<td>0.56</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.52</td>
<td>0.47</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0.21</td>
<td>0.25</td>
<td>0.14</td>
</tr>
<tr>
<td>R2</td>
<td>12</td>
<td>0.72</td>
<td>0.73</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.17</td>
<td>0.14</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>0.15</td>
<td>0.20</td>
<td>0.14</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.04</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.79</td>
<td>0.80</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>0.38</td>
<td>0.40</td>
<td>0.37</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td>0.44</td>
<td>0.43</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Table 5.3: DAG-SVM+MRF results with predefined regions.

The results of each of these experiments can be seen in Tables 5.1 and 5.2 respectively. The two correlation metrics show difference aspects of the performance of the two regression techniques.

In terms of pure correlation, as measured by the PCC, the best results were achieved with SVRs when applied to the full face image features, with mean PCC of 0.45. In fact, when applied to the face region features, the trend is reversed, with the DAG-SVMs significantly outperforming the SVRs when measured by PCC performance (0.44 vs 0.41).

However, the ICC metric gives a more consistent relative performance, with the DAG-SVMs achieving the highest correlation in both cases, and the highest mean ICC, of 0.42, being achieved with DAG-SVM classifiers when applied to the face region features. As the ICC can be taken as a better overall indicator of performance than the PCC, this means that in absolute terms, the DAG-SVMs method is able to predict the AU intensities more accurately in both cases.

However, both measures taken together show interesting behaviour in the two regression techniques. When the face is divided into two regions, the performance of the SVRs is lower than when features are chosen from the entire face image. However, this trend is reversed in the case of the DAG-SVMs, as the better performance is then achieved when features are only taken from the face regions. This is the case, even though the same
5.4. Experiments

<table>
<thead>
<tr>
<th>Region</th>
<th>AU</th>
<th>PCC</th>
<th>ICC</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>R3</td>
<td>1</td>
<td>0.57</td>
<td>0.57</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.52</td>
<td>0.51</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.63</td>
<td>0.63</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.43</td>
<td>0.41</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.46</td>
<td>0.41</td>
<td>0.66</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0.22</td>
<td>0.17</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.72</td>
<td>0.70</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.19</td>
<td>0.14</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>0.14</td>
<td>0.13</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.03</td>
<td>0.02</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.75</td>
<td>0.73</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>0.39</td>
<td>0.38</td>
<td>0.75</td>
</tr>
<tr>
<td>Mean</td>
<td>0.42</td>
<td>0.40</td>
<td>0.39</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Table 5.4: DAG-SVM+MRF with selected regions.

features are employed for both regression techniques. This apparent contradiction can be explained by considering the very different operation of the two regression techniques - DAG-SVMs rely on accurate classification by each of the individual binary SVM classifiers trained only on two intensity classes at a time, whereas SVRs aim to fit a function to all intensity points. As a result, the feature choice can greatly affect performance of each method in different ways, and in this case demonstrates that the predefined regions are better suited to the intensity classification method, whereas the features available in the selected region method allow for better performance by the SVRs.

5.4.2 Model Performance

In order to evaluate the performance of the MRF tree model approach, the regressors were applied as input, and the performance compared between this raw input and the intensities predicted by the chosen tree models. Trees were constructed for each experiment from the AUs that could be present in each of the face regions.

Regressor Comparison To investigate which regressor is best suited for employment as input to the MRF method, experiments were conducted using each of the two feature selection methods. The predefined regions included only a subset of AUs in trees for two regions: upper face and lower face. Upper face trees were created which included: 1, 2, 4, 5, 6 and 9. Lower face trees: 12, 15, 17, 20, 25, and 26. For the first experiment,
features were also selected only from the appropriate face region. The selected regions method operated on the entire face image and involved building trees including all 12 of the possible AUs.

The results of employing DAG-SVMs as input with the selected regions and predefined can be seen in Tables 5.3 and 5.4 respectively. Similarly the results of applying SVRs as inputs can be seen in 5.5 and 5.6. The latter two tables demonstrate how the MRF trees show a significant improvement over the input in the majority of AUs and on average. The mean performance shows a large improvement, particularly in the case of the predefined

<table>
<thead>
<tr>
<th>Region</th>
<th>AU</th>
<th>PCC</th>
<th>ICC</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SVR</td>
<td>MRF</td>
<td>SVR</td>
</tr>
<tr>
<td>R1</td>
<td>1</td>
<td>0.51</td>
<td>0.56</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.49</td>
<td>0.59</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.55</td>
<td>0.64</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.39</td>
<td>0.44</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.51</td>
<td>0.53</td>
<td>0.43</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0.20</td>
<td>0.24</td>
<td>0.16</td>
</tr>
<tr>
<td>R2</td>
<td>12</td>
<td>0.76</td>
<td>0.78</td>
<td>0.66</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.10</td>
<td>0.15</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>0.26</td>
<td>0.33</td>
<td>0.21</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.07</td>
<td>0.13</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.72</td>
<td>0.76</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>0.37</td>
<td>0.42</td>
<td>0.33</td>
</tr>
<tr>
<td>Mean</td>
<td>0.41</td>
<td>0.47</td>
<td>0.35</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Table 5.5: SVR+MRF results with predefined regions.

<table>
<thead>
<tr>
<th>Region</th>
<th>AU</th>
<th>PCC</th>
<th>ICC</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SVR</td>
<td>MRF</td>
<td>SVR</td>
</tr>
<tr>
<td>R3</td>
<td>1</td>
<td>0.53</td>
<td>0.56</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.56</td>
<td>0.63</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.53</td>
<td>0.55</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.53</td>
<td>0.56</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.52</td>
<td>0.55</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0.27</td>
<td>0.39</td>
<td>0.22</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.74</td>
<td>0.79</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.17</td>
<td>0.16</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>0.24</td>
<td>0.37</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.10</td>
<td>0.11</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.72</td>
<td>0.77</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>0.44</td>
<td>0.42</td>
<td>0.38</td>
</tr>
<tr>
<td>Mean</td>
<td>0.45</td>
<td>0.49</td>
<td>0.38</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Table 5.6: SVR+MRF results with selected regions.
5.4. Experiments

<table>
<thead>
<tr>
<th>Region</th>
<th>AU</th>
<th>PCC</th>
<th></th>
<th>ICC</th>
<th></th>
<th>RMSE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SVR</td>
<td>MRF</td>
<td>SVR</td>
<td>MRF</td>
<td>SVR</td>
<td>MRF</td>
</tr>
<tr>
<td>R1</td>
<td>1</td>
<td>0.53</td>
<td>0.55</td>
<td>0.50</td>
<td>0.54</td>
<td>0.66</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.56</td>
<td>0.61</td>
<td>0.47</td>
<td>0.60</td>
<td>0.57</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.53</td>
<td>0.59</td>
<td>0.42</td>
<td>0.54</td>
<td>1.10</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.53</td>
<td>0.54</td>
<td>0.51</td>
<td>0.51</td>
<td>0.24</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.52</td>
<td>0.52</td>
<td>0.45</td>
<td>0.45</td>
<td>0.68</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0.27</td>
<td>0.38</td>
<td>0.22</td>
<td>0.36</td>
<td>0.57</td>
<td>0.55</td>
</tr>
<tr>
<td>R2</td>
<td>12</td>
<td>0.74</td>
<td>0.77</td>
<td>0.63</td>
<td>0.76</td>
<td>0.81</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.17</td>
<td>0.16</td>
<td>0.16</td>
<td>0.14</td>
<td>0.43</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>0.24</td>
<td>0.34</td>
<td>0.20</td>
<td>0.29</td>
<td>0.61</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.10</td>
<td>0.13</td>
<td>0.08</td>
<td>0.11</td>
<td>0.43</td>
<td>0.43</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.72</td>
<td>0.77</td>
<td>0.57</td>
<td>0.76</td>
<td>1.27</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>0.44</td>
<td>0.50</td>
<td>0.38</td>
<td>0.47</td>
<td>0.79</td>
<td>0.66</td>
</tr>
<tr>
<td>Mean</td>
<td>0.45</td>
<td>0.49</td>
<td>0.38</td>
<td>0.46</td>
<td>0.68</td>
<td>0.61</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.7: SVR+MRF results with selected region features, but predefined region specific trees.

regions, in all three performance measures PCC (0.47 vs 0.41), ICC (0.44 vs 0.35) and RMSE (0.60 vs 0.77).

However, this pattern is not seen when the DAG-SVM approach is applied as input, as seen in Tables 5.3 and 5.4. In both cases, the MRF performance does not show any improvement on average, and for the majority of AUs there is little or no improvement seen. The reason for this could be because of the nature of the DAG-SVM output. Though able to predict individual AU intensities well, the combination of many binary classifiers may give a less consistent set of outputs across the AUs for different intensity combinations, which gives an signal that unsuitable for training models which rely on regular regressor patterns for different intensity combinations, as in this case. As a result, for the remainder of this section only the SVRs will be employed as input for the model.

**Region Selection Method** In order to determine the optimal region selection method, the results when the SVRs were applied as input were examined, as seen in Tables 5.5 and 5.6. These results demonstrate the much larger improvement seen when trees are constructed with fewer AUs over that seen in the selected region method (that builds trees from all 12 AUs). In the predefined case, an improvement is seen in all three measures, PCC, ICC and RMSE, for every AU, and the mean results also demonstrate a significant improvement by every measure (PCC 0.47 vs 0.41, ICC 0.44 vs 0.35 and RMSE 0.60 vs 0.77). An improvement is also seen in the selected region case, but it is a
5. Facial Action Unit Intensity Estimation

Figure 5.5: Examples comparing the predictions given by regression and the MRF approach. GT - Ground Truth, SVR - Support Vector Regressor, S+M - SVR+Markov Random Field. (a) AU1 subject SN008 (b) AU2 subject SN004 (c) AU4 subject SN004 (d) AU5 subject SN023 (e) AU9 subject SN023 (f) AU12 subject SN008 (g) AU17 subject SN008 (h) AU25 subject SN032.
5.4. Experiments

### Analysis

In order to provide further analysis of the results, Table 5.8 is included to show the percentage of the frames for which each tree is chosen by each of the methods. The three SVR+MRF methods outlined above are compared here:

- **SVR+MRF1**: Predefined regions, with region specific trees.
- **SVR+MRF2**: Selected regions, with entire face image trees.
- **SVR+MRF3**: Selected region features, with region specific trees.

<table>
<thead>
<tr>
<th>Method</th>
<th>Percentage of Frames Tree Chosen</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR+MRF1†</td>
<td>26.8 24.3 29.7 5.3 12.1 1.7 7.8 1.7 8.5 4.5 74.3 3.2</td>
</tr>
<tr>
<td>SVR+MRF2</td>
<td>5.9 5.5 9.9 1.6 0.1 0.2 1.9 0.4 2.1 0.5 71.1 0.8</td>
</tr>
<tr>
<td>SVR+MRF3†</td>
<td>25.9 21.5 35.4 4.7 10.0 2.5 7.3 1.9 8.4 4.9 74.9 2.6</td>
</tr>
</tbody>
</table>

Table 5.8: The percentage of frames for which each tree is chosen. † In these methods the AUs are divided into two regions so each half of table sums to 100%

A smaller increase for most AUs, and in this case there is no improvement seen for a small number of AUs. This could be because the added complexity of the larger trees allows for more error during the inference process.

However, as the SVR results were shown to be higher when allowing feature selection from the entire face image, a final experiment was conducted, which combined these superior features with the face region trees. The results of this experiment can be seen in Table 5.7. Again, the face region specific trees demonstrate a significant improvement over the raw SVR results, as measured by every metric, achieving 0.49 mean PCC and 0.45 mean ICC, compared to 0.45 and 0.38 respectively for the SVRs alone. A significant improvement is also seen for the majority of AUs as measured by all three metrics, and a much higher performance achieved for AU 26 than with either of the previous methods (PCC 0.50 vs 0.42 and ICC 0.47 vs 0.38). Generally, however, this method only produces a comparable performance on average to the predefined method (Table 5.5), hence both methods produce the highest AU intensity estimation performance as measured by the PCC and ICC metrics.
5. Facial Action Unit Intensity Estimation

This table demonstrates how for all three methods the trees are chosen in similar proportions, apart from Method 2 where the tree for AU 25 is chosen for a much larger percentage of frames.

In addition, examples of the comparative performance for a number of AUs are included in Fig. 5.5. These examples demonstrate particular instances when the combinational aspect of the MRF can clearly be seen to improve performance, over that possible from appearance components alone.

From the results, it can be seen that there are a number of AUs for which the MRF method gives a particularly large increase in performance. For example, AU 2 (Outer Brow Raise) is clearly improved dramatically in all three cases. Looking at the ICC measure in particular, the MRF method achieves a much higher performance in all three cases (0.58, 0.62 and 0.61) than for AU 1 (Inner Brow Raise) (0.55, 0.55, 0.54), despite much lower comparative SVR performance for these two AUs (0.40, 0.47, 0.47 vs 0.45, 0.50, 0.50). These two AUs are very correlated, as they each consist of the upwards movement of a part of the eyebrows, and as a result will generally have the same intensity whenever a surprised or shocked expression is present. The results suggest that this method is able to harness these large correlations to improve the estimation for AU 2, and that this could be because the tree with AU 1 as root is chosen a larger percentage of the time than the one corresponding to 2, as can be seen in Table 5.8. An example of the result of how much improvement can be gained for AU 2 can be seen in Fig. 5.5b. Here, there are clear cases, around frames 1300 – 1400 and 4200 – 4500, where the intensity prediction cannot be explained by the underlying regressor values alone, and thus have been influenced by the regressor values for other AUs, which in this case is likely to be AU.

The other figures in Fig. 5.5 demonstrate other examples of when the predicted intensities have clearly been influenced by other AU parts in the tree. Particular examples are in Fig. 5.5g, where at around frame 4550 the correct intensity is predicted, despite a considerable drop in the actual regressor value for this AU at that point, and Fig. 5.5h, where though the predictions generally follow a similar trend to the regressor, there are examples of both the predicted intensity increasing when the regressor reducing and the reverse.
5.4.3 Comparison with Previous Work

Finally, the performance of this method is compared to the current state-of-the-art work in AU intensity estimation on this database. As previous authors have generally reported only the PCC and RMSE scores of their methods, comparisons are conducted by these metrics only. Three previous works are compared against, each of which has investigated AU intensity estimation on this dataset: Using NMF transformed features in order to train linear SVRs [64], and training CCRFs [63], and CCNFs [8] on this problem.

The PCC and RMSE scores for each are displayed in Figs. 5.6a and 5.6b respectively, along with the results of the previous work methods applied to the DISFA databases. The results quoted here are taken from [8].

These results show how the SVR+MRF approach achieves superior performance to all previous methods on all of the upper face AUs (1, 2, 4, 5, 6, and 9) and on AU 12 (smile). It also achieves comparable, or superior, performance than the majority of previous methods on a number of other AUs 17, 20, 25, and 26. Only on AU 15 does this method fail to achieve comparable PCC scores than the previous work. But on average, this method in all three cases outperforms two of the previous methods (NMF+SVR [64] and CCRF [63, 8]), and achieves the same average performance as the most recent work employing a CCNF method [8].

5.4.4 Summary and Discussion

These experiments have shown a number of important results. DAG-SVMs were shown to be well suited for use as regressors in this problem, achieving superior performance to SVRs when trained on both sets of features as measured by the ICC metric, though SVRs did achieve the highest PCC score on selected regions. The best results, as measured by the ICC, were produced when DAG-SVMs were applied to features chosen from predefined regions.

The results then demonstrated how harnessing the correlations between AUs can considerably improve intensity estimation over that possible from regression alone, when SVRs are employed as inputs. This technique was shown to improve intensity estimation for
5. Facial Action Unit Intensity Estimation

Figure 5.6: Comparison of previous techniques with the SVR+MRF method proposed here. Previous methods: NMF+SVR [64], CCRF [63], CCNF [8]. Proposed methods: SVR+MRF1 - predefined regions, SVR+MRF2 - selected regions, SVR+MRF3 - combination. (a) PCC performance (b) RMSE performance.
all AUs when employed with predefined regions, and the majority of AUs with selected regions and the combined method. It also achieved performance that was superior to the state-of-the-art methods previously presented in a number of AUs, as well as achieving comparable performance on average.

The experiments conducted here demonstrate a number of limitations to this approach. Not all regressors are suitable as input to this model approach. DAG-SVMs, though able to outperform SVRs when employed alone, do not give a signal that can be harnessed successfully by the MRF models. This shows that the approach requires a stable, predictable, signal, across the full set of AUs, to work. This may be due to the simplicity of the individual facial region models employed, and particularly the fact that a generative approach was employed here. Using discriminative methods, such as Structured Support Vector Machines (SSVMs), may be a way to produce a more robust method, that could also improve on the results presented here.

The use of 2D data may be one reason that intensity estimation for the majority of lower face AUs is low, as compared to the performance on the upper face AUs. As demonstrated in Chapter 3, 3D features give superior performance on AUs such as 15, 17 and 20, where the shapes of points around the mouth are particularly important for accurate recognition. The use of these features, in combination with this intensity estimation method, would therefore be expected to improve performance across the the lower face AUs. The combination of both 2D and 3D features simultaneously would also be expected to give an improvement in performance in generally, as was demonstrated in that same Chapter.

The temporal information available in the video sequences was exploited in only the simplest way possible - smoothing through application of a moving average. However, this gave a large increase in performance in all cases, demonstrating that the dynamics of AUs are an important factor in analysis. Future work could look at utilising this information in more intelligent ways, in order to improve on the estimation possible from static frames alone.

Finally, accurate AU intensity estimation would pave the way for estimation of intensity
of a wide range of expressions, and detection of complex emotional states, for many applications. This higher-level analysis is where the strength of the AU approach comes into its own, as it provides a comprehensive coding of the full motion of the face, that is not tied to any subset of expressions of particular application. Hence, the benefits of this coding, once fully accurate, would be immeasurable.

5.5 Conclusions

In this chapter, the task of AU intensity estimation has been explored. For this purpose, different regression techniques, including the first application of DAG-SVMs to this task, were explored and compared. DAG-SVMs were shown to generally outperform the more traditional SVR approach. Different region selection methods, prior to feature selection, were also explored.

Then, in order to exploit correlations between AUs in a face region for better estimation, a selection of MRF tree structures were employed, each of which represents when one AU is dominant in the region. The trees were constructed from subsets of the training labels, according to AUs that demonstrated the highest correlation, and then the appropriate parameters were calculated for each tree based on the regression outputs when tested on the training set. Message passing was then applied for inference on each tree, and the most likely tree chosen at each frame in order to determine the full set of intensity predictions for the AUs in the region. This technique was successfully shown to outperform the regression results alone when applied to SVR inputs, and achieved a superior performance to the previous work in this area on a number of AUs, as well as the same performance on average.
6.1 Contributions

This thesis provides a number of contributions to the field of facial expression analysis, particularly in the areas of 3D facial geometry and facial AU analysis, which are summarised here.

Chapter 3 introduced a range of novel 3D binary pattern features, which were successfully applied to the problem of facial AU detection. A number of representations of the 3D facial geometries were exploited, the depth map, facial normals, APDI and APCI, to create a set of features based on binary pattern methods that had previously been successfully applied to 2D data. Different methods for feature fusion were explored, in order to harness the complementary nature of a number of the features, and combining 2D and 3D binary pattern features was shown to achieve optimal performance. The features were parameter optimised, and the performance fully examined through comparison with the equivalent 2D features, other previously proposed methods, and cross-database testing. A novel region selection method was also introduced, to improve the computation cost of the method, whilst maintaining performance.

Chapter 4 outlined a novel method for automatic dynamic analysis of the six basic expressions in 3D data. Motion-based features were captured in each pair of dimensions, spatial and time, through a quad-tree method that focussed attention on areas of highest motion. The features were then subjected to feature selection employed in GB classification.
as inputs to temporal models of the full expression sequence (neutral-onset-apex-offset). A comparison was conducted with the equivalent 2D method and the 3D approach shown to be superior. In addition, the temporal analysis was shown to improve recognition rates over the classifier outputs alone.

Chapter 5 introduced methods for AU intensity estimation. Firstly, techniques for regression were explored, including the use of region selection for choosing optimal features. Secondly, the correlations between AUs were harnessed for improved estimation through exploitation of a MRF tree framework. Different face regions were represented by a number of possible tree structures, each the maximum intensity AU in the region as root. The appropriate tree could then be automatically chosen as the most likely tree with best intensity combination estimation for the full set of AUs.

6.2 Findings

This thesis has developed techniques in a number of areas of facial expression analysis, that have thus far received little attention. The experiments conducted have also extended knowledge of these areas.

It was demonstrated in Chapter 3 that 3D binary pattern features are extremely beneficial and complementary to 2D features in the task of facial AU detection. In particular, 3D features tend to outperform 2D features on a number of the lower face AUs, where the shape of the mouth is especially important for distinguishing between a number of different AUs. Combining the two feature types, however, led to the best overall performance, and so it would appear that including both datatypes is the way forward in this area. Chapter 4 confirmed that 3D data is also superior to 2D images for a number of expressions. Finally, Chapter 5 demonstrated that the lower face AUs are those for which intensity estimation is least accurate, which could be due to the use of 2D features alone. In summary, the findings of all of these chapters show that the use of both modalities is required for optimal performance in any AU or expression analysis application.

Exploiting both temporal and spatial structural information was shown to give benefits over static or individual information alone. Chapter 4 demonstrated how the use of
the HMM successfully smoothed the output of the GB classifiers to give a much more accurate representation of the full expression presence. Chapter 5 then demonstrated how the spatial correlations between AUs can be harnessed to improve intensity estimation over regression alone.

Finally, region selection was shown to be hugely beneficial for AU detection and intensity estimation. The ability to maintain or improve performance, whilst reducing computation cost considerably, through employment of a simple bulk feature selection technique is important for building future systems for AU analysis.

6.3 Limitations and Future Work

There are a number of areas in which the methods introduced here could be further improved or extended, particularly in the area of facial AU intensity estimation. A number areas are focussed on here: the difficulties of working with 3D and 4D data, the inclusion of AU intensity dynamics in order to improve estimation performance, the extension of these techniques to 3D spontaneous facial expression data, and finally, the potential use of deep learning methods in future work on 2D and 3D facial expression analysis.

6.3.1 3D Data Storage and Processing

The use of 3D and 4D data is still limited to a large extent by the size of the data, and time required for processing. 3D databases are generally an order of magnitude larger than 2D databases, due to the need to store the data as 3D meshes, as well as texture information. This can make acquisition, storing, and transferral of such data difficult, which impedes progress in this area. In addition, the computational expense of processing each frame mesh can also make analysis of data of this kind difficult and time consuming. As hard drives and computational power increases these issues will become less important and make 3D/4D facial expression acquisitions and analysis on a large scale more feasible.
6. Conclusions

6.3.2 Intensity Estimation Dynamics

Facial expressions are known to demonstrate a number of dynamic patterns, which can be harnessed as demonstrated in Chapter 4. Particularly the speed of the onset and offset of AUs is known to hold important information on the emotional state [3]. Therefore, as well as expecting correlations between AUs within the face, which were successfully exploited in Chapter 5, it is also expected that combining this with the temporal correlations between frames would further improve the estimation of facial AU intensities.

To harness this information, the temporal information could be modelled using a similar approach to that adopted in Chapter 4, but adapted to AUs, employed on intensities rather than temporal segments, and combined with the spatial frame models, in order to fully exploit both correlations at once. To do this, the full set of predicted intensity combinations from all trees could be combined for each frame, to form a set of emission probabilities for the intensities of each AU. This could then be employed as input to a set of HMMs, one for each AU, with the states set as the different possible intensities.

It would be expected that this could greatly improve on the results possible from single frames alone, due to the large amount of temporal information available in AU data. For example, it is physically impossible for frames containing a high intensity AU to not be preceded and followed by at least some presence of that AU. Additionally, temporal correlations between different AUs would also be expected, which can also be exploited.

6.3.3 3D Spontaneous Expression Analysis

This thesis has mainly examined the problem of facial expression analysis in 3D data, with Chapters 3 and 4 both focussed on particular aspects of this area. However, it was not possible to investigate 3D AU intensity estimation due to to the lack of a suitable dataset being available in time. In order to explore real correlations between AUs in 3D, spontaneous 4D data, annotated with AU intensities, is required. For the first time, the BP4D-Spontaneous database fulfils this need. It contains 4D sequences of spontaneous examples of the six basic expressions plus a number of other expressions, with full AU intensity codings provided for each sequence. Now that this database has been released,
work on AU intensity estimation in 3D data is possible.

In order to undertake this work, several aspects from each of the three chapters in this thesis could be combined, to build a full 4D AU intensity estimation system. The binary pattern based features proposed in Chapter 3 could be exploited on the facial geometries in this new database, and employed as input to the spatial-temporal MRF-HMM model. It would be expected that this would improve performance on a number of the lower face AUs, that were shown to be difficult to detect with 2D features.

6.3.4 Deep Learning as an Alternative Approach

Facial expression analysis built on AU detection of intensity estimation, naturally appears to lend itself to the new ideas in deep learning [6]. These methods aim to employ ANNs for building hierarchical structures that model higher-level abstractions. In this case, one obvious approach would be to aim to model the relationships between images, AUs and expressions. These ideas are currently gaining interest, with a number of researchers working on them. It is therefore possible that in the near future the field could be moved along by this area of research, with new possibilities opening in expression analysis and
the inference of emotional states.

6.4 Summary

In summary, this thesis has developed techniques for 4D facial analysis through binary pattern features, and temporal and structural modelling. The work presented here paves the way for a full 4D facial expression analysis system, based on AU intensity estimation from structural and temporal information. The techniques developed here could be widely employed to build systems suitable for a number of facial analysis applications, such as human-computer interaction, deception detection, pain detection, or medical diagnosis.
Publications


Appendix

In this appendix the full results from the basic and filter based features in Chapter 3 are included for reference, along with more extensive discussion of these results for individual AUs. Tables 1 and 2, show the full results for the basic and filter based features respectively.

Basic Features

These results show that, of the basic 3D feature types, the new 3D features all perform at least comparably to LDBPs on average, though the majority are narrowly outperformed by the 2DLBPs. All of the features are outperformed on average by the LNPs, though only marginally in the case of the LABP_C and the LABP_D+LNBP_OA fusion which achieve average results of 95.3 and 95.1 respectively. However, the individual AU results demonstrate more interesting information about the discriminative abilities of each of the feature types and their complementary nature. For a number of AUs, particularly the upper face AUs, the 2DLBPs outperform or equal all of the 3D features. This is what might be

<table>
<thead>
<tr>
<th>AU</th>
<th>2DLBP</th>
<th>LDBP</th>
<th>LNP</th>
<th>LABP_D</th>
<th>LABP_C</th>
<th>LNBP_C</th>
<th>LNBP_T</th>
<th>LNBP_T2</th>
<th>AB_D+NB_OA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>97.9 ±2.2</td>
<td>97.7 ±5.0</td>
<td>97.9 ±2.5</td>
<td>96.1 ±2.7</td>
<td>96.3 ±3.0</td>
<td>93.0 ±3.0</td>
<td>95.9 ±2.3</td>
<td>96.2 ±2.9</td>
<td>96.4 ±2.3</td>
</tr>
<tr>
<td>2</td>
<td>99.4 ±0.7</td>
<td>97.2 ±1.6</td>
<td>98.7 ±0.7</td>
<td>98.4 ±0.9</td>
<td>99.0 ±0.7</td>
<td>95.1 ±3.0</td>
<td>97.9 ±1.2</td>
<td>98.3 ±1.0</td>
<td>98.2 ±1.0</td>
</tr>
<tr>
<td>4</td>
<td>98.2 ±1.4</td>
<td>96.9 ±2.5</td>
<td>98.2 ±3.0</td>
<td>98.4 ±2.8</td>
<td>96.7 ±2.6</td>
<td>95.9 ±3.3</td>
<td>96.4 ±3.2</td>
<td>96.4 ±2.6</td>
<td>98.4 ±1.9</td>
</tr>
<tr>
<td>5</td>
<td>94.8 ±2.5</td>
<td>92.7 ±2.8</td>
<td>94.0 ±4.4</td>
<td>92.5 ±4.7</td>
<td>94.4 ±2.8</td>
<td>90.6 ±3.6</td>
<td>93.4 ±3.2</td>
<td>92.1 ±3.8</td>
<td>93.9 ±3.1</td>
</tr>
<tr>
<td>7</td>
<td>95.4 ±2.7</td>
<td>90.9 ±7.9</td>
<td>93.0 ±4.9</td>
<td>91.7 ±4.1</td>
<td>91.7 ±4.9</td>
<td>87.5 ±8.8</td>
<td>85.8 ±10.9</td>
<td>85.2 ±8.7</td>
<td>91.5 ±4.6</td>
</tr>
<tr>
<td>8</td>
<td>97.0 ±1.7</td>
<td>93.9 ±2.5</td>
<td>96.5 ±2.4</td>
<td>95.5 ±2.6</td>
<td>96.1 ±2.1</td>
<td>96.0 ±1.9</td>
<td>94.9 ±2.4</td>
<td>95.9 ±2.1</td>
<td>97.0 ±1.3</td>
</tr>
<tr>
<td>38</td>
<td>77.3 ±7.1</td>
<td>77.2 ±8.1</td>
<td>83.6 ±6.8</td>
<td>69.7 ±1.0</td>
<td>81.6 ±5.6</td>
<td>73.8 ±16.0</td>
<td>83.4 ±5.9</td>
<td>75.8 ±6.1</td>
<td>75.8 ±12.5</td>
</tr>
<tr>
<td>43</td>
<td>98.8 ±1.4</td>
<td>97.5 ±2.2</td>
<td>98.4 ±1.8</td>
<td>96.8 ±2.5</td>
<td>97.3 ±2.3</td>
<td>97.6 ±1.9</td>
<td>97.7 ±2.1</td>
<td>96.9 ±2.2</td>
<td>97.7 ±2.2</td>
</tr>
<tr>
<td>9</td>
<td>99.3 ±0.7</td>
<td>99.4 ±0.6</td>
<td>99.7 ±0.2</td>
<td>99.7 ±0.4</td>
<td>99.7 ±0.3</td>
<td>99.5 ±0.3</td>
<td>99.6 ±0.5</td>
<td>99.3 ±0.7</td>
<td>99.7 ±0.3</td>
</tr>
<tr>
<td>10</td>
<td>96.7 ±2.3</td>
<td>94.8 ±3.8</td>
<td>96.5 ±3.2</td>
<td>95.1 ±3.2</td>
<td>96.7 ±2.5</td>
<td>97.0 ±2.1</td>
<td>95.7 ±3.0</td>
<td>94.6 ±3.0</td>
<td>96.7 ±2.5</td>
</tr>
<tr>
<td>12</td>
<td>99.0 ±0.6</td>
<td>97.0 ±2.0</td>
<td>98.7 ±1.0</td>
<td>98.6 ±0.9</td>
<td>98.4 ±1.2</td>
<td>97.6 ±1.8</td>
<td>97.9 ±1.6</td>
<td>98.1 ±1.3</td>
<td>98.6 ±0.9</td>
</tr>
<tr>
<td>R12</td>
<td>95.6 ±6.7</td>
<td>93.8 ±10.3</td>
<td>96.5 ±5.0</td>
<td>94.3 ±6.6</td>
<td>96.7 ±3.4</td>
<td>95.9 ±5.4</td>
<td>95.4 ±4.8</td>
<td>95.4 ±3.6</td>
<td>95.6 ±6.2</td>
</tr>
<tr>
<td>L12</td>
<td>93.5 ±3.6</td>
<td>95.5 ±4.2</td>
<td>92.9 ±6.9</td>
<td>92.8 ±4.9</td>
<td>89.9 ±6.3</td>
<td>95.8 ±1.8</td>
<td>95.4 ±2.3</td>
<td>94.7 ±2.0</td>
<td>96.9 ±2.1</td>
</tr>
<tr>
<td>14</td>
<td>94.9 ±2.5</td>
<td>93.0 ±3.9</td>
<td>94.7 ±4.2</td>
<td>95.1 ±4.5</td>
<td>96.9 ±2.0</td>
<td>95.4 ±3.7</td>
<td>92.7 ±4.0</td>
<td>95.1 ±2.6</td>
<td>96.5 ±4.1</td>
</tr>
<tr>
<td>15</td>
<td>92.3 ±5.1</td>
<td>83.6 ±4.7</td>
<td>96.0 ±3.4</td>
<td>94.3 ±4.6</td>
<td>95.0 ±1.9</td>
<td>97.6 ±7.4</td>
<td>91.7 ±4.6</td>
<td>91.2 ±5.7</td>
<td>94.7 ±5.9</td>
</tr>
<tr>
<td>16</td>
<td>84.5 ±4.7</td>
<td>89.1 ±3.2</td>
<td>90.4 ±2.3</td>
<td>85.7 ±4.8</td>
<td>90.9 ±2.2</td>
<td>88.6 ±3.3</td>
<td>88.6 ±5.5</td>
<td>90.4 ±4.2</td>
<td>88.8 ±3.0</td>
</tr>
<tr>
<td>17</td>
<td>92.0 ±5.5</td>
<td>90.1 ±5.1</td>
<td>94.9 ±3.3</td>
<td>90.8 ±5.9</td>
<td>94.5 ±3.5</td>
<td>86.1 ±1.0</td>
<td>93.1 ±5.1</td>
<td>91.3 ±4.7</td>
<td>92.5 ±4.7</td>
</tr>
<tr>
<td>18</td>
<td>96.7 ±1.5</td>
<td>97.1 ±1.0</td>
<td>97.5 ±1.2</td>
<td>97.1 ±1.3</td>
<td>96.8 ±1.7</td>
<td>97.6 ±1.9</td>
<td>97.0 ±1.3</td>
<td>98.3 ±1.1</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>96.2 ±4.0</td>
<td>93.3 ±6.2</td>
<td>99.4 ±0.4</td>
<td>98.7 ±1.5</td>
<td>98.0 ±1.9</td>
<td>91.4 ±5.7</td>
<td>95.6 ±3.9</td>
<td>94.6 ±4.0</td>
<td>97.4 ±1.7</td>
</tr>
<tr>
<td>22</td>
<td>94.2 ±4.5</td>
<td>96.5 ±5.5</td>
<td>97.5 ±3.6</td>
<td>95.9 ±6.8</td>
<td>97.8 ±2.9</td>
<td>94.4 ±5.9</td>
<td>97.2 ±4.7</td>
<td>96.7 ±4.8</td>
<td>97.1 ±3.3</td>
</tr>
<tr>
<td>23</td>
<td>86.8 ±10.7</td>
<td>86.2 ±10.3</td>
<td>91.6 ±5.8</td>
<td>84.9 ±6.7</td>
<td>89.3 ±7.6</td>
<td>81.9 ±10.0</td>
<td>76.1 ±7.6</td>
<td>84.0 ±6.5</td>
<td>86.0 ±8.0</td>
</tr>
<tr>
<td>24</td>
<td>93.4 ±3.1</td>
<td>94.0 ±4.1</td>
<td>96.2 ±2.7</td>
<td>94.7 ±2.6</td>
<td>95.9 ±2.2</td>
<td>92.8 ±3.0</td>
<td>94.9 ±1.9</td>
<td>95.4 ±1.7</td>
<td>96.3 ±2.5</td>
</tr>
<tr>
<td>25</td>
<td>97.5 ±1.7</td>
<td>99.3 ±0.6</td>
<td>99.6 ±0.4</td>
<td>98.9 ±0.9</td>
<td>99.6 ±0.4</td>
<td>99.4 ±0.6</td>
<td>99.3 ±0.6</td>
<td>99.5 ±0.5</td>
<td>99.6 ±0.5</td>
</tr>
<tr>
<td>26</td>
<td>86.2 ±4.6</td>
<td>87.1 ±5.1</td>
<td>88.3 ±5.0</td>
<td>87.3 ±5.3</td>
<td>89.3 ±4.7</td>
<td>87.1 ±5.0</td>
<td>88.0 ±6.0</td>
<td>89.5 ±4.8</td>
<td>88.5 ±5.0</td>
</tr>
<tr>
<td>27</td>
<td>98.6 ±0.9</td>
<td>97.4 ±2.5</td>
<td>98.4 ±1.2</td>
<td>98.0 ±1.3</td>
<td>98.5 ±1.1</td>
<td>95.6 ±3.5</td>
<td>97.5 ±2.0</td>
<td>98.1 ±1.2</td>
<td>97.9 ±1.6</td>
</tr>
<tr>
<td>28</td>
<td>98.8 ±1.8</td>
<td>95.9 ±6.0</td>
<td>98.6 ±2.7</td>
<td>97.7 ±2.0</td>
<td>98.5 ±2.1</td>
<td>98.6 ±1.4</td>
<td>97.4 ±4.2</td>
<td>95.1 ±4.2</td>
<td>98.7 ±1.5</td>
</tr>
<tr>
<td>34</td>
<td>98.6 ±1.2</td>
<td>96.7 ±4.4</td>
<td>98.6 ±2.0</td>
<td>97.2 ±2.6</td>
<td>97.2 ±4.1</td>
<td>95.4 ±1.6</td>
<td>94.9 ±2.0</td>
<td>94.2 ±2.0</td>
<td>98.7 ±0.6</td>
</tr>
</tbody>
</table>

Table 1: ROC AuC results ($\mu \pm \sigma$) for all AUs when employing basic feature types. Comparison features are in italics.
expected in the case of AUs whose distinguishing aspects are lines in the skin such as 1 and 2 (which usually produce furrows in the forehead), 2D/3D examples of the latter shown in Figs. 3.11a and 3.11g) and 6 (which produces crow’s feet on the outside corners of the eyes), examples shown in Figs. 3.11b and 3.11h). In addition, AUs 5 and 43 both involve the eyelids, which are not captured cleanly in the 3D images in the Bosphorus database. as can be seen in the examples shown in Figs. 3.11i-3.11j. This can mean that these AUs are easier to detect from the contrast between an open and closed eye in texture. (as seen in Figs. 3.11c-3.11d) than from the geometry change.

However, there are a number of AUs, such as 14, 15, 16, 20, 22, 23 and 28, where one or more of the 3D features significantly outperform the 2D features. This is also in line with expectations, as these AUs involve movements such as large changes in relative depth, or
subtle shape changes, particularly around the mouth. For example, protruding lips are important for distinguishing AU 16, and are more clear in the facial geometry shown in 3.11s than the 2D image 3.11m. And the dimples important for 14 are shown more clearly in 3.11k than 3.11e. Finally, stretching or bunching of the lips, important for AUs such as 17 or 20, are also easier to detect from the facial geometry than the texture alone, as can be seen in the contrasting 2D/3D examples in Fig. 3.11. Although LNPs achieve the highest overall result, there are a large number of AUs for which they are outperformed by one or more of the new features. These include 4 (LABP_D), 10, 14, and 22 (LABP_C), L12 and 18 (LABP_D+LNBP_OA), and 26 (LNBP_TA2). In addition, a comparable result is achieved in a number of other AUs. As all of these features save LNBP_TA2 are significantly shorter in length than LNPs, this demonstrates how a more compact representation of the normal information is still able to capture a large amount of the discriminant behaviour necessary for AU detection. These results also show that, though there is a marginal benefit on some AUs from the 2D LNBP_TA method, the 1D method achieves a similar, or better result on the majority of AUs, with a far shorter descriptor. The complementary nature of the 2D and 3D features is in line with the results that were reported in [143]. In that work, they also reported that 2D features outperformed on upper face AUs, but 3D features were superior on lower face AUs.

Filter-based Features

The filter-based features generally outperform the basic features, as can be seen in Table 2. These results show that LDMBPs and LDGBPs produce the highest mean results, with a slight improvement over 2DLGBPs achieved with these feature types on average. However, again, the individual AUs show more interesting discriminative behaviour. Here a similar pattern can be observed as in the basic feature case, where 2D features are superior for upper face AUs, but the trend is reversed in the lower face AUs. In addition, it is clear from these results that the LDMBPs features yield the highest result the majority of the time, and achieve the highest mean result of 96.3. LDGBPs also achieve a comparable mean score (96.0), and similarly many comparable individual results, however the fact that they do equally well is an interesting outcome due to the fact that the monogenic method requires a much shorter feature descriptor as it does not need to be applied at a range
Appendix

of orientations. LDPQs also achieve a comparable result to either Gabor or Monogenic depth features for a large number of AUs, though they are outperformed on average. In the case of the azimuthal features, LAGBPs generally outperform LAMBPs, achieving a higher mean result. They also outperform the depth features for a small number of AUs such as 2, 10 and 20. LAPQs do not perform well, suggesting that the LPQ algorithm is not beneficial for extracting further information from the APDI representation.
Bibliography

[1] 3DMD 4D Capture, May 2011. 44


[14] C. Beumier and M. Acheroy. 3d facial surface acquisition by structured light. In In International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, 1999. 44


167
Bibliography


Bibliography


[92] Dimensional Imaging Ltd. www.di4d.com. 44


173
Bibliography


174


Bibliography


177


