Face Recognition in Low Resolution Using a 3D Morphable Model

Pouria Mortazavian

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

March 2013

© Pouria Mortazavian 2013
Summary

Biometric person identification has been an active research area in the recent decades. Among the many biometric features, such as finger prints, face, and voice; the human face offers particular advantages such as the ability to recognise an individual from a distance, and without their cooperation.

Although face recognition using 2D images taken under controlled conditions has reached high performance rates, its performance declines under non-cooperative scenarios such as surveillance using CCTV cameras where the scene can have arbitrary illumination conditions and the subject can have arbitrary pose with respect to the camera and be at a far distance.

The focus of this thesis is on the problem of recognising individuals from a 2D facial image with low resolution and arbitrary pose and illumination. We investigate the use of 3D information in order to boost the performance of 2D face recognition in such scenarios. A 3D Morphable Face Model is used to extract 3D shape and facial texture information from a 2D low-resolution facial image with arbitrary pose and illumination. To this end, the 3D model is fitted to the input image using a novel low-resolution fitting algorithm proposed in this thesis. It is shown that the fitting algorithm is able to extract reliable 3D shape and texture information across a large range of variations in pose and illumination.

It is shown, through extensive experimental evaluation, that the model parameters obtained using our fitting algorithm are reliable enough to be directly used for face recognition in low-resolution under varying poses and illuminations.

Furthermore, we propose a novel approach to using 3D information in order to enhance the low-resolution facial texture. More specifically, we propose a 3D-assisted facial texture super-resolution framework which uses the 3D information extracted from an LR image to map the facial texture to a shape- and pose-normalised domain. The facial texture is then enhanced by applying texture super-resolution in this domain. Through this procedure, a high-resolution estimate of the facial texture is obtained which can then be used to render the face in a normalised pose and illumination and with high-resolution texture. It is shown that this procedure can inject relevant and discriminative high-resolution information to the facial texture thereby boosting the performance of a conventional 2D face recognition engine.

Key words: Face Recognition, 3D Morphable Model, Fitting, Resolution, Super-resolution.

Email: p.mortazavian@surrey.ac.uk
WWW: http://www.surrey.ac.uk/cvssp/
Acknowledgements

I would like to express my sincerest gratitude to my supervisors Professor Josef Kittler and Dr. William Christmas for their valuable guidance and constructive comments. They have been a source of inspiration for me, both professionally and personally, and I truly feel honoured to have had the privilege of working with them.

I would also like to thank my friends and colleagues at the Centre for Vision, Speech, and Signal Processing (CVSSP) for their support and for various constructive discussions from which I have definitely learnt a lot. In particular, I would like to thank Dr. Rafael Tena for the helpful discussions during the initial stages of this research.

Furthermore, I would like to thank Mr. David Banks and Mr. Barry Flannaghan, directors at InSync Technology Ltd., for their support and understanding during the final writing stages of this thesis.

Further gratitude is due to the Overseas Research Students Awards Scheme (ORSAS) for partly funding my research.

Finally, and on a different level, I would like to express my deepest appreciation and love to my parents, Tahereh and Mostafa, and my brother, Parham, for their continuous and unconditional love and support throughout the years.
Contents

List of Symbols ix

1 Introduction 1
  1.1 Face Recognition .................................................. 2
    1.1.1 Some Terminology ............................................ 3
    1.1.2 Current Challenges ........................................... 3
    Pose: ................................................................. 5
    Illumination: ..................................................... 5
    Expression: ....................................................... 5
    Resolution: ....................................................... 5
  1.2 Motivation .......................................................... 6
  1.3 Thesis Outline ...................................................... 7
  1.4 Contributions ...................................................... 8
  1.5 List of Publications ............................................... 8

2 3D Morphable Model for Unconstrained Face Recognition 9
  2.1 Introduction ........................................................ 9
  2.2 Building a 3D Morphable Model .................................... 10
    2.2.1 Registered 3D Faces .......................................... 11
    2.2.2 PCA-Based Face Space ....................................... 12
    2.2.3 Segmented Model .............................................. 17
  2.3 3DMM for Image Synthesis ........................................... 17
    2.3.1 Shape projection ............................................. 18
    2.3.2 Inverse Shape Projection .................................... 18
    2.3.3 Illumination and Colour Transformation ...................... 20
## Contents

Colour transformation: ............................................. 22
Occlusion and shadow: ........................................... 23

2.3.4 Image synthesis .......................................... 23

2.4 Model Fitting to a 2D Image ................................ 24
  2.4.1 Stochastic Newton Optimisation .................... 24
  2.4.2 Multi-Feature Fitting ................................ 27
      Pixel Colour ............................................. 28
      Edges ..................................................... 28
      Anchor Points ......................................... 31
      Specular Highlights .................................. 31
      Gaussian Prior ......................................... 31
      Texture Constraint ..................................... 32
      Light Regularisation .................................. 32

Our Implementation ........................................... 33
      Light Regularisation: ................................ 33
      Colour Transformation: ................................. 33
      Parameters and Features: ............................... 33
      Use of Anchor Points: ................................ 34
      Edge Cost: ............................................... 34
      Translation in Z: ....................................... 34
      Illumination Model: .................................... 34

2.5 Texture Extraction ........................................ 35

2.6 Face Recognition Using a 3D Morphable Face Model .... 37
      Image-Based Approach: ................................ 37
      Model-Based Approach: ................................ 37

2.7 Discussion and Conclusions ............................ 37

3 A Review of Low-Resolution Face Recognition Techniques 41
  3.1 Introduction .............................................. 41
  3.2 Super-Resolution ......................................... 43
      3.2.1 Observation Model ............................... 44
      3.2.2 Reconstruction-Based Methods ................. 46
3.2.3 Example-Based Methods .............................................. 50
3.2.4 Face Super-Resolution ............................................. 51
3.3 Direct HR-LR Matching ............................................... 55
3.4 Model-Based Approaches ............................................. 57
3.5 Discussion and conclusions ......................................... 62

4 Resolution-Aware Fitting of 3DMM on LR Images .................. 67
4.1 Introduction ............................................................ 67
4.2 The Resolution Problem .............................................. 68
4.3 A Low-Resolution Image Formation Model ....................... 70
  4.3.1 Centroid-based Linear Texture Model ......................... 71
  4.3.2 Low-Resolution Image Synthesis ............................. 72
      Shape Projection ................................................. 73
      Texture Illumination and Colour Transform ................... 73
  4.3.3 The Low-Resolution Image Formation Model ................. 74
  4.3.4 The Reference Resolution ...................................... 75
4.4 Fitting Algorithm for Low-Resolution ............................ 77
  4.4.1 Pixel Colour Cost ............................................. 77
  4.4.2 Edge Cost .................................................... 79
      Pyramid Formation: ............................................ 80
      Multi-Resolution and Multi-Parameter Edge Analysis: ....... 80
      Cost Surface Fusion: ........................................... 81
  4.4.3 Robust Estimation ............................................. 82
  4.4.4 Low-Resolution Model Fitting ................................ 83
4.5 Experimental Evaluation ............................................ 84
4.6 Discussion and Conclusions ....................................... 93

5 3DMM for Face Recognition in Low Resolution ..................... 97
5.1 Introduction .......................................................... 97
5.2 Using Model Parameters for Face Recognition .................... 97
5.3 Face Identification .................................................. 99
  5.3.1 Finding the Reference Resolution ............................ 100
5.3.2 The gallery ........................................ 101
5.3.3 Illumination-Invariant LR Face Identification .............. 102
5.3.4 Pose-Invariant LR Face Identification ...................... 105
5.3.5 Pose- and Illumination-Invariant Face Identification ....... 107
5.4 Face Verification ......................................... 108
  5.4.1 LDA-Based Face Verification ............................ 110
  5.4.2 Experimental Setup and Results ......................... 111
5.5 Summary and Conclusion .................................. 114

6  A 3D-Assisted Facial Texture Super-Resolution Framework ...... 115
  6.1 Introduction ............................................. 115
  6.2 Proposed Framework ...................................... 116
    6.2.1 Formulation ......................................... 116
    6.2.2 Results .............................................. 122
    6.2.3 Further Improvements ................................. 122
    Model Fitting and Texture Extraction ...................... 126
    Illumination Normalisation ................................ 127
    Texture Map Correction ................................. 128
  6.3 3D-Assisted Face Recognition in Low Resolution............... 130
  6.4 Discussion .............................................. 132
  6.5 Conclusions ............................................. 137

7 Conclusions and Future Work ................................ 139
  7.1 Future work ............................................ 140
    7.1.1 Efficiency .......................................... 140
    7.1.2 Multi-Frame Model Fitting ............................ 141
    7.1.3 Illumination Estimation ............................... 142
    7.1.4 Automation .......................................... 143
  7.2 A Plea to the Community ................................ 144

A Result of 3D-Assisted Facial Texture Super-Resolution .......... 145

Bibliography .................................................... 151
Acronyms and Mathematical Notation

Acronyms

3DMM 3D Morphable Face Model
AAM  Active Appearance Model
ASM  Active Shape Model
CDT Chamfer Distance Transform
DSF  Down-Sampling Factor
EER  Equal Error Rate
FAR  False Acceptance Rate
FRR  False Rejection Rate
HR   High Resolution
HTER Half Total Error Rate
IBL  Image-Based Likelihood
IBP  Iterative Back Projection
ICIA Inverse Compositional Image Alignment
IMDR Inverse Multi-Resolution Dense Registration
LDA  Linear Discriminant Analysis
LR   Low Resolution
MAP  Maximum Aposteriori Estimation
MDS  Multidimensional Scaling
MFF  Multi-Feature Fitting
MI   Mutual Information
ML   Maximum Likelihood Estimation
MRF  Markov Random Field
NC   Normalised Correlation
NN   Nearest Neighbour
PCA  Principal Components Analysis
PDF  Probability Distribution Function
PSF  Point Spread Function
PSNR Peak Signal to Noise Ratio
SDA  Simultaneous Discriminant Analysis
SNO  Stochastic Newton Optimisation
SR   Super Resolution
Mathematical Notation

Chapter 2

$x, y, z$ Cartesian coordinates of a 3D point ............................................ 11
$r, g, b$ Colour values of a point ................................................................. 11
$S^\text{raw}_k$ Raw shape vector of the $k^{th}$ 3D face scan, before registration ........... 11
$T^\text{raw}_k$ Raw texture vector of the $k^{th}$ 3D face scan, before registration ........... 11
$S_k$ Shape Vector of the $k^{th}$ face scan, after registration .......................... 12
$T_k$ Texture Vector of the $k^{th}$ face scan, after registration .......................... 12
$N_v$ Number of vertices of the generic model .............................................. 12
$M$ Number of face scans used for training the model .................................... 14
$D_S$ Dimensionality of the shape PCA space .............................................. 15
$D_T$ Dimensionality of the texture PCA space ............................................. 15
$S_i^e$ $i^{th}$ shape eigenvector ................................................................. 15
$T_i^e$ $i^{th}$ texture eigenvector ................................................................. 15
$\alpha$ Vector of model shape parameters ................................................. 15
$\beta$ Vector of model texture parameters ................................................. 15
$\sigma^2_{S,i}$ Variance of the 3DMM's $i^{th}$ shape component ............................ 15
$\sigma^2_{T,i}$ Variance of the 3DMM's $i^{th}$ texture component ............................ 15
$s_i$ 3D object-centred coordinates of the $i^{th}$ vertex .................................. 18
$w_i$ 3D world coordinates of the $i^{th}$ vertex .............................................. 18
$p_i$ 2D image coordinates of vertex $i$ ....................................................... 18
$(p_i, q_i)$ 2D image coordinates of vertex $i$ ................................................... 18
$p(\mathbf{x}; \tau)$ Vector valued shape projection function .................................. 18
$\theta_x, \theta_y, \theta_z$ Rotation angles (projection parameters) ............................ 18
$t_{w,x}, t_{w,y}, t_{w,z}$ Translations in the world coordinates (projection parameters) ........ 18
$f$ Camera focal length (projection parameter) ............................................ 18
## Contents

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau )</td>
<td>Vector of projection parameters</td>
<td>18</td>
</tr>
<tr>
<td>( p^{-1}(p, q; \tau) )</td>
<td>Vector valued inverse shape projection function</td>
<td>18</td>
</tr>
<tr>
<td>( o )</td>
<td>Function composition</td>
<td>18</td>
</tr>
<tr>
<td>( L_r^a, L_g^a, L_b^a )</td>
<td>RGB components of the ambient light (Illumination parameters)</td>
<td>20</td>
</tr>
<tr>
<td>( L_r^d, L_g^d, L_b^d )</td>
<td>RGB components of the directional light (Illumination parameters)</td>
<td>20</td>
</tr>
<tr>
<td>( t_i )</td>
<td>Vector of RGB albedo values of the ( i^{th} ) vertex</td>
<td>20</td>
</tr>
<tr>
<td>( t'_i )</td>
<td>Illuminated texture of the ( i^{th} ) vertex</td>
<td>20</td>
</tr>
<tr>
<td>( n_i^{v, w} )</td>
<td>Normal vector of the ( i^{th} ) vertex (world coordinates)</td>
<td>20</td>
</tr>
<tr>
<td>( d )</td>
<td>Light direction vector</td>
<td>20</td>
</tr>
<tr>
<td>( r_i )</td>
<td>Light reflection direction vector for the ( i^{th} ) vertex</td>
<td>20</td>
</tr>
<tr>
<td>( v_i )</td>
<td>Viewing direction vector for the ( i^{th} ) vertex</td>
<td>20</td>
</tr>
<tr>
<td>( n_j^{v, o} )</td>
<td>Normal vector of the ( j^{th} ) triangle (object-centred coordinates)</td>
<td>21</td>
</tr>
<tr>
<td>( g_r, g_g, g_b )</td>
<td>R, G and B channel gains (Illumination parameters)</td>
<td>22</td>
</tr>
<tr>
<td>( o_r, o_g, o_b )</td>
<td>R, G and B channel offsets (Illumination parameters)</td>
<td>22</td>
</tr>
<tr>
<td>( c )</td>
<td>Colour correction contrast (Illumination parameter)</td>
<td>22</td>
</tr>
<tr>
<td>( t_i^C )</td>
<td>Illuminated and colour-corrected texture of the ( i^{th} ) vertex</td>
<td>22</td>
</tr>
<tr>
<td>( t^C(.) )</td>
<td>Vector valued illumination function</td>
<td>23</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Vector of illumination parameters</td>
<td>23</td>
</tr>
<tr>
<td>( \mu )</td>
<td>Vector of model parameters (shape and texture)</td>
<td>24</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Vector of rendering parameters (projection and illumination)</td>
<td>24</td>
</tr>
</tbody>
</table>

### Chapter 3

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h )</td>
<td>A high-resolution image, represented as a vector</td>
<td>44</td>
</tr>
<tr>
<td>( l )</td>
<td>A low-resolution image, represented as a vector</td>
<td>44</td>
</tr>
<tr>
<td>( D_k )</td>
<td>Down-sampling operation associated with the ( k^{th} ) LR observation</td>
<td>44</td>
</tr>
<tr>
<td>( B_k )</td>
<td>Blurring operation associated with the ( k^{th} ) LR observation</td>
<td>44</td>
</tr>
<tr>
<td>( W_k )</td>
<td>Warping operation associated with the ( k^{th} ) LR observation</td>
<td>44</td>
</tr>
<tr>
<td>( n_k )</td>
<td>Additive noise associated with the ( k^{th} ) LR observation</td>
<td>44</td>
</tr>
</tbody>
</table>
M_k Unified observation matrix associated with the $k^{th}$ LR observation ..... 45
H A high-resolution image ................................................. 46
L A low-resolution image ................................................... 46
$(m, n)$ Pixel coordinates in a low-resolution image ......................... 46
$(p, q)$ Pixel coordinates in a high-resolution image ......................... 46

Chapter 4

m 2D integer pixel coordinates of an (LR) image pixel ......................... 68
u 2D real valued coordinates of an image pixel ................................ 68
E Continuous irradiance light field that reaches the image plane ............. 68
PSF Point Spread function of a camera ........................................ 68
A Area of a pixel ................................................................. 68
t^C Illuminated and colour-corrected model texture ................................. 68
x 3D object-centred coordinates of a point on the model’s surface ............ 68
p^{-1}(.) Vector valued inverse shape projection function ........................ 68
° Function composition .......................................................... 69
t_i Texture of vertex i (vertex-based model) ...................................... 71
\hat{t}_k Texture of triangle k (centroid-based model) ............................... 71
\hat{t}_{ave} Average texture of triangle k in the centroid-based model ............ 71
\hat{t}^{e}_{i;k} $i^{th}$ basis vector of the centroid-based texture model, corresponding to triangle k .... 71
T_{ave} Average texture vector (centroid-based model) ............................. 72
\hat{T}_i Vector of all texture basis vectors (centroid-based model) ................ 72
N_i Number of triangles in the centroid-based texture model .................... 72
\hat{T} Facial texture described by the centroid-based model ....................... 72
\hat{p}(x; \rho) Vector-valued, weak-perspective shape projection function ......... 73
\hat{p}^{-1}(x; \rho) Inverse of the vector-valued, weak-perspective shape projection function .... 73
h Half-way vector (Blinn-Phong illumination model) ............................... 73
E_{c}^{LRF} LR Pixel colour cost term .............................................. 77
## Contents

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>Multi-resolution, multi-parameter edge cost surface</td>
<td>81</td>
</tr>
<tr>
<td>$E_e^{LRF}$</td>
<td>LR edge cost term</td>
<td>82</td>
</tr>
</tbody>
</table>

### Chapter 5

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\zeta$</td>
<td>Identity Vector</td>
<td>98</td>
</tr>
<tr>
<td>$\hat{\alpha}^g$</td>
<td>Normalised global shape vector</td>
<td>98</td>
</tr>
<tr>
<td>$\hat{\alpha}^s_i$</td>
<td>Normalised shape vector corresponding to the $i^{th}$ segment</td>
<td>98</td>
</tr>
<tr>
<td>$\hat{\alpha}^b$</td>
<td>Normalised shape vector corresponding to the blended model</td>
<td>98</td>
</tr>
<tr>
<td>$\hat{\beta}^g$</td>
<td>Normalised global texture vector</td>
<td>98</td>
</tr>
<tr>
<td>$\hat{\beta}^s_i$</td>
<td>Normalised texture vector corresponding to the $i^{th}$ segment</td>
<td>98</td>
</tr>
<tr>
<td>$\hat{\beta}^b$</td>
<td>Normalised texture vector corresponding to the blended model</td>
<td>98</td>
</tr>
<tr>
<td>$n_\alpha$</td>
<td>Number of model shape parameters</td>
<td>98</td>
</tr>
<tr>
<td>$\sigma_{S,i}$</td>
<td>Standard deviation of the $i^{th}$ shape component</td>
<td>98</td>
</tr>
<tr>
<td>$n_\beta$</td>
<td>Number of model texture parameters</td>
<td>98</td>
</tr>
<tr>
<td>$\sigma_{T,i}$</td>
<td>Standard deviation of the $i^{th}$ texture component</td>
<td>98</td>
</tr>
<tr>
<td>$\nu^k$</td>
<td>One of 12 (normalised) parameter vectors obtained by model fitting</td>
<td>110</td>
</tr>
<tr>
<td>$W^k$</td>
<td>LDA transform corresponding to $\nu^k$</td>
<td>110</td>
</tr>
<tr>
<td>$\chi^k$</td>
<td>Projection of $\nu^k$ to corresponding LDA space</td>
<td>110</td>
</tr>
<tr>
<td>$s_{a,b}^k$</td>
<td>Similarity score between the $k^{th}$ parameter vectors of samples a &amp; b</td>
<td>110</td>
</tr>
<tr>
<td>$s_{a,b}$</td>
<td>Similarity score between samples a &amp; b</td>
<td>110</td>
</tr>
<tr>
<td>$T$</td>
<td>Operating threshold of a face verification system</td>
<td>112</td>
</tr>
<tr>
<td>$T_{EER}$</td>
<td>EER threshold of a face verification system</td>
<td>112</td>
</tr>
</tbody>
</table>

### Chapter 6

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>HR texture map</td>
<td>116</td>
</tr>
<tr>
<td>$L$</td>
<td>LR image</td>
<td>117</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>( \mu )</td>
<td>The set of 3DMM parameters (shape and texture)</td>
<td>117</td>
</tr>
<tr>
<td>( \rho )</td>
<td>The set of rendering parameters (projection and illumination)</td>
<td>117</td>
</tr>
<tr>
<td>( \delta(.) )</td>
<td>Dirac delta function</td>
<td>117</td>
</tr>
<tr>
<td>( t )</td>
<td>LR texture map</td>
<td>117</td>
</tr>
<tr>
<td>( (p,q) )</td>
<td>Texel coordinates in an HR texture map</td>
<td>119</td>
</tr>
<tr>
<td>( (m,n) )</td>
<td>Texel coordinates in an LR texture map</td>
<td>119</td>
</tr>
<tr>
<td>( w )</td>
<td>Down-sampling factor</td>
<td>119</td>
</tr>
<tr>
<td>( \sigma^2 )</td>
<td>Covariance of the texel noise distribution</td>
<td>119</td>
</tr>
<tr>
<td>( \mathcal{G}_l(I) )</td>
<td>Level ( l ) of the Gaussian pyramid of image ( I )</td>
<td>120</td>
</tr>
<tr>
<td>( \mathcal{L}_l(I) )</td>
<td>Level ( l ) of the Laplacian pyramid of image ( I )</td>
<td>120</td>
</tr>
<tr>
<td>( \mathcal{H}_l(I) )</td>
<td>First horizontal derivative of ( \mathcal{G}_l(I) )</td>
<td>120</td>
</tr>
<tr>
<td>( \mathcal{V}_l(I) )</td>
<td>First vertical derivative of ( \mathcal{G}_l(I) )</td>
<td>120</td>
</tr>
<tr>
<td>( \mathcal{H}^2_l(I) )</td>
<td>Second horizontal derivative of ( \mathcal{G}_l(I) )</td>
<td>120</td>
</tr>
<tr>
<td>( \mathcal{V}^2_l(I) )</td>
<td>Second vertical derivative of ( \mathcal{G}_l(I) )</td>
<td>120</td>
</tr>
<tr>
<td>( \text{PS}_l(I)(m,n) )</td>
<td>( l )th level parent structure vector of image ( I ), at pixel ( (m,n) )</td>
<td>120</td>
</tr>
<tr>
<td>( \tilde{\mathcal{H}}_0(I) )</td>
<td>Predicted horizontal derivative of image ( I ), at level 0</td>
<td>121</td>
</tr>
<tr>
<td>( \tilde{\mathcal{V}}_0(I) )</td>
<td>Predicted vertical derivative of image ( I ), at level 0</td>
<td>121</td>
</tr>
<tr>
<td>( \sigma^2_{\mathcal{H}} )</td>
<td>Covariance of horizontal gradient prediction error distribution.</td>
<td>121</td>
</tr>
<tr>
<td>( \sigma^2_{\mathcal{V}} )</td>
<td>Covariance of vertical gradient prediction error distribution.</td>
<td>121</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

“I’m Spartacus!”

The iconic phrase above is from the classic movie “Spartacus” by Stanley Kubrik, based on the life of the historic figure Spartacus and the events of the Third Servile War. After the army of rebel slaves is defeated by the Roman army, the recaptured slaves are offered leniency by the victorious Romans under the single condition that they identify their leader, Spartacus. The rebels reject the offer, each proclaiming himself to be Spartacus, thus protecting their leader by creating a huge number of false claims so as to render the Romans unable of identifying the one true Spartacus.

Historic fact or dramatic fiction, this event represents an example of identity fraud where a person falsely claims the identity of another. In this particularly unusual case, it was a heroic act with morally sound intentions, but that is rarely the case in real life\(^1\)! In reality, identity fraud is often an ill-intentioned act and is usually a part of a more serious criminal operation.

Governmental agencies, banks, and insurance companies regularly warn citizens and clients about the implications and dangers of identity fraud.

The UK Fraud Prevention Service (CIFAS) defines identity fraud as:

*The creation or adoption of a fictitious or false identity to facilitate illegal or fraudulent activity. This usually involves the use of stolen or forged identity documents such as a passport or driving licence to obtain goods or services by deception. Includes cases of false identity, identity theft, impersonation of the deceased, facility takeover and other impersonations.*\[^93\]

CIFAS recorded more than 120 thousand cases of identity fraud in 2012, which accounts for about 50% of all cases of fraud recorded to their National Fraud Database during 2012 \[^94\].

\(^1\)See also Section 7.2 for a discussion.
Preventing identity fraud requires a reliable method of verifying a person’s identity. To date, various means of person identity verification have been used. These can roughly be categorised into three main groups of methods:

**Knowledge-based** systems determine identity of a person based on *what they know*, e.g. a password, a Personal Identification Number (PIN), address, date of birth.

**Token-based** systems determine identity of a person based on *what they possess*, e.g. an ID card, passport, driving licence.

**Biometric** systems determine identity of a person based on a specific *physiological* or *behavioural* characteristic, e.g. fingerprint, facial appearance, gait, voice.

The first two categories of methods are by far the most common means of person identification. However, since they are not based on an inherent attribute of an individual, they suffer from a number of disadvantages and are susceptible to fraudulent misuse. An identity token can be lost or stolen, the knowledge (passwords etc.) can be forgotten or wrongfully acquired by an unauthorised imposter. Interestingly enough, such disadvantages have manifested themselves in CIFAS’s definition of identity fraud given in Page 1, where it mentions that identity fraud “usually involves the use of stolen or forged identity documents”.

On the other hand, biometric systems in general, and physiological biometric systems in particular, are inherently more reliable in differentiating between an authorised person and an imposter since most physiological and behavioural characteristics are specific to a given person and hard to mimic. Hence, biometric person identification systems have gained considerable attention both in academic research and in the industry. A recent industrial report by the International Biometrics Group, LLC (IBG), provides annual revenue projections and forecasts that the industry revenues will rise from 3.4 billion US dollars to roughly 9.4 billion US dollars within the 6 years from 2009 to 2014 [53].

### 1.1 Face Recognition

Among various biometric modalities, face recognition earns special attention. Faces are probably the most common biometric attributes used by humans for recognition. Face recognition offers a non-intrusive biometric method and ideally requires no user cooperation. Hietmeyer [85] compared different biometric attributes in terms of compatibility with a Machine Readable Travel Documents (MRTD) [1] system based on a number of evaluation factors such as enrollment, renewal, and public perception. In his study, the face features scored highest among the 6 attributes\(^2\) considered. According to the Biometrics Market and Industry Report (BMIR) by IBG, face recognition systems account for 11.4% of the total biometrics market in 2009, second only to fingerprints.

Considering the above, it comes as no surprise that automatic face recognition is one of the most active areas of research in computer vision and pattern recognition. Applications are not limited to security and access control, but also span a range of other areas such as human-machine interaction and automatic organisation of family photos.

\(^2\)The attributes considered by Hietmeyer include Face, Finger, Hand, Voice, Eye, and Signature.
1.1. Face Recognition

1.1.1 Some Terminology

In general, the automatic face recognition problem can be considered in two different scenarios in terms of the user’s interaction with the system: (1) cooperative user scenarios and (2) non-cooperative user scenarios. In the first case, the user is willing to be cooperative by presenting their face to the system in a way that facilitates automatic recognition (e.g. in frontal pose and with a neutral expression); a typical example of such a scenario is in access control systems. In the second case, the user is not necessarily aware of being identified; a typical case is the surveillance by CCTV. In such cases the user may have arbitrary pose and distance with respect to the camera. In terms of distance between the user and the camera, near-field (typically, less than 1m) face recognition in a cooperative scenario is the least difficult problem whereas far-field non-cooperative scenarios pose the most challenging problem.

Face recognition is a pattern recognition problem which operates by processing facial samples in the form of 2D visible-light images, 3D scans, or images beyond the visible spectrum (e.g. Near-Infrared (NIR)). In an enrollment stage, a set of individuals are enrolled in the system by presenting one or more facial samples of each individual. These individuals will then be known to the system and are referred to as clients. The set of client samples used for enrollment is referred to as the gallery set. Then, in a test stage, a different set of samples is presented to the system for recognition. This set is referred to as the probe set.

The term face recognition is an umbrella term referring to two closely related tasks, namely face verification and face identification.

Face verification is a one-to-one matching problem where an individual presents a facial sample to the system and claims the identity of one of the clients. The task is then to verify whether this claim is true, i.e. the person is a genuine client; or false, i.e. the person is an imposter.

Face identification, on the other hand, is a one-to-many matching problem where a person presents a face sample to the system and the task is to determine the identity of the person. Two scenarios can be considered in evaluating a face identification system, the so-called closed-set identification scenario is the case where the probe set used for evaluating the performance of the system consists only of clients who are known by the system. A more general scenario, on the other hand, is the open-set identification scenario where the probe set is allowed to include samples of people not known to the system, i.e. imposters, in addition to clients.

1.1.2 Current Challenges

Early research on automatic face recognition started in the 1960s [64]. The first face recognition system was developed by Takeo Kanade in his PhD thesis, in 1973 [57]. Within the four decades that have passed, development of powerful and affordable computing systems has resulted in an enormous boost in application fields of automatic image analysis and computer vision such as medical image analysis, media and entertainment, human-machine interaction, and biometric identification. Face recognition has been no exception. Many milestones have been set since the pioneering
work of Kanade by various researchers. Perhaps the most notable early works were the Eigenface method \cite{103}, which applied Principal Components Analysis (PCA) to the problem, and the Fisherface method which achieved higher accuracy using Linear Discriminant Analysis (LDA) \cite{12, 39}.

More recent breakthroughs were made using local appearance-based features such as Local Binary Patterns (LBP) \cite{3}, Scale-Invariant Feature Transform (SIFT) \cite{69}, and Histogram of Oriented Gradients (HOG) \cite{31}.

As a result of years of research and advances in face recognition, the technology has currently reached a certain level of maturity to the extent that in a cooperative scenario where factors such as lighting, pose, expression, and distance can be controlled, current automatic face recognition systems can outperform humans \cite{64}. The Face Recognition Vendor Test 2002 (FRVT2002) compares commercially available face recognition systems and shows that the performance of the best systems under such constrained conditions is comparable to that of finger print recognition systems \cite{83}.

However, accurate and reliable non-cooperative face recognition under unconstrained imaging conditions is still an open area of research and the technology has not yet matured sufficiently in order to make its way into every day use. Many challenges still need to be addressed before face recognition systems can be deployed as reliable biometric person identification systems for unconstrained scenarios, especially in sensitive applications such as security. The specific challenges vary among systems depending on the type of facial samples used by the system. For instance, high cost and limitations in 3D acquisition systems pose challenges to 3D-based systems whereas they are not typically significant problems for a 2D image-based system which in turn faces a challenge in handling different subject poses. Other factors such as facial expression could pose a challenge for all systems.

In the following, we restrict ourselves to the case where the facial samples presented to the system for enrollment and for test are 2D visible-light images and discuss some of the main challenges for such systems.

A facial image conveys many pieces of information. Facial shape, facial texture, the subject’s gender, age, facial expression, pose, distance to camera, direction of gaze and information about scene illumination are among the main pieces of information conveyed by a facial image. From a recognition point of view, the different sources of variation in an image can be divided into two main groups: (1) intrinsic sources of variation are those that are intrinsic to the identity of the subject; and (2) extrinsic sources of variation are those that are independent of the identity. Among the various sources of variation named above, we consider only facial shape and texture to be intrinsic factors, directly related to identity, and consider all the rest as extrinsic sources of variation \footnote{One could argue that facial expressions are not, strictly speaking, an extrinsic factor as they could convey some information about a person’s identity.}.

Each of the various intrinsic and extrinsic pieces of information originates from a separate source and could vary rather independently of the others. Ideally, a face recognition system should be able to differentiate between the intrinsic and extrinsic
factors, thus extracting from an image only that part of the information which is directly relevant to identity.

However, in a 2D image all various kinds of intrinsic and extrinsic information are irreversibly combined by virtue of the image acquisition process. Hence, any discrepancy between the gallery and probe images in terms of the extrinsic factors introduces unwanted noise in the system and can negatively affect the performance of a 2D recognition engine.

Among the different types of extrinsic information conveyed by a face image, variations in some factors prove to be more damaging to the performance of a recognition system than others. For instance, a change in a person’s gaze direction may not affect the performance of a recognition system as much as variation in illumination can. In the following, we discuss the most challenging extrinsic factors, variations of which can significantly degrade the performance of a 2D face recognition engine.

**Pose:** Variations in facial pose can greatly change the appearance of a face in a 2D image. A 2D facial image represents a projection of a 3D object (the face) to a 2D plane. Consequently, the face is seen differently from different view points. Different parts of the face become visible or occluded, and their relative spatial positions vary, depending on the relative position of the camera with respect to the face.

**Illumination:** A 2D image is created as a result of interaction between light and the face. The light reflected by the face is captured by an imaging device. Hence, the appearance of a human face in an image can vary dramatically due to variations in the illumination conditions. Moreover, a directional light can cast shadows on the face thereby creating strong contours on the face. It has been shown \[2, 112\] that differences in facial appearance due to illumination can be larger than differences between individuals.

**Expression:** Humans are capable of showing a wide range of various different facial expressions depending on the internal emotional state, intention, or social interactions. Depending on the type and intensity of the facial expression, facial features can vary significantly. Hence, facial expressions are among the main sources of variation in the appearance of a face in a 2D image and can have a significant impact on the performance of any face recognition system.

**Resolution:** In an unconstrained scenario, the captured face may be of low resolution due to the large distance between the subject and the camera and/or the camera characteristics. Low image resolution can lead to loss of descriptive and intrinsic facial information conveyed by the image, thereby affecting the performance of a face recognition system. Different studies have shown that a face recognition engine’s performance can degrade as a result of low resolution and large distance between the subject and the camera \[106, 18\].
1.2 Motivation

We reviewed the main challenges of a 2D image-based face recognition system in the previous section. There has been a sizable body of previous work on addressing each of the individual challenges separately. In comparison, the problem of face recognition across a combination of these factors simultaneously has received relatively little attention. However, the ultimate goal of face recognition is to be able to recognise faces in the presence of any arbitrary combination of such extrinsic factors.

Our main focus in this work is on the problem of face recognition in low resolution (LR). However, instead of considering this problem in isolation of other challenging problems, we aim to provide a way towards low-resolution face recognition in the presence of variations in other extrinsic factors such as pose and illumination.

As mentioned in the previous section, a major source of the challenges for a 2D face recognition systems is that, during image acquisition, various kinds of information are combined. Separating the various pieces of information conveyed by a 2D image is an ill-posed problem and requires prior knowledge about the human face.

As we will show through an extensive review of the low-resolution face recognition literature in Chapter 3, the most successful methods applied to the problem of LR face recognition use prior knowledge about a (high-resolution) human face in order to reconstruct and extract high-resolution (HR) information from an LR image. However, most of the current approaches to LR face recognition are prone to limitations in recognising faces of previously unseen poses and illuminations.

On the other hand, one of the most successful approaches proposed to date which uses prior knowledge of human face characteristics for the purpose of 2D facial image analysis is the 3D Morphable Face Model (3DMM), proposed by Blanz and Vetter [19]. The 3DMM provides a framework to model various kinds of intrinsic and extrinsic information explicitly, separately and independently. The 3DMM has been successfully applied to the problem of 2D face recognition in the presence of pose and illumination variation [86, 88] and has shown promising potential for expression-invariant face recognition [4]. However, its application is limited when it comes to low-resolution images\(^4\).

Considering the above discussion, the main motivation for this work was to try and combine the merits of 3DMM with those of the most powerful methods in the LR face analysis literature. The 3DMM offers valuable object-specific information about the human face. Inspired by ideas from the LR face analysis literature, we aim to appropriately use this prior knowledge in order to extract and separate various pieces of intrinsic and extrinsic information from a LR facial image. The objective is to provide a possible solution to the problem of LR face recognition in the presence of variations in other extrinsic factors, in particular pose and illumination, thereby taking a small step in the path towards truly unconstrained face recognition.

\(^4\)See Section 4.2 for a discussion
1.3 Thesis Outline

This thesis is organised in 7 chapters. The current chapter gives an introduction to the terminology and main challenges of face recognition as a biometric personal identification technique. It also outlines the main motivation for the presented work and provides a list of contributions and publications derived from this work. Chapter 2 presents a review of the 3D morphable face model and its application to face recognition under varying poses and illuminations. We present a review of the related literature on building a 3D morphable model and the main approaches to face analysis through model fitting.

In Chapter 3, we review the challenge of low-resolution (LR) face recognition and some of the approaches that are used to address it. We identify four main categories of approaches: (1) training in LR; (2) using HR reconstruction (super-resolution) to enhance the probe image prior to recognition; (3) model-based approaches; and (4) direct matching of HR and LR faces. Through a critical analysis of the reviewed approaches, we argue that the present methods suffer from serious limitations in the analysis of facial images of previously unseen poses and illuminations.

Given this necessary background, in Chapter 4 we will make the link between super-resolution (SR) and 3DMM and show how the underlying ideas used in SR can be transferred to the domain of 3D morphable models and utilised to extend the use of 3DMM to LR face analysis. More specifically, we propose an LR imaging model for generating LR images given the 3DMM parameters and a set of rendering parameters. We then use this imaging model in developing an algorithm for fitting a 3DMM to LR images. Experimental evaluation of this algorithm confirms that our proposed fitting approach can successfully fit the 3DMM to LR images and that the fitted parameters are similar to those that would have been obtained if the input image was HR.

We then move on to the application of the 3DMM in LR face recognition in Chapter 5. By using the 3DMM parameters directly for comparison of gallery and probe images in a series of identification experiments, we show that the proposed fitting algorithm provides sufficient accuracy for face recognition under varying poses and/or illuminations over a range of different resolutions. In addition, we propose a novel LDA-based methodology for using the 3DMM parameters in face recognition and evaluate this approach in a face verification scenario.

In Chapter 6, we pursue a different approach to using the 3DMM in LR face analysis. More specifically, we present two alternative approaches for using the 3D information provided by the 3DMM in order to aid 2D LR face analysis. First, we present a modular 3D-assisted facial texture super-resolution framework which uses the 3D information to map facial texture to a pose- and illumination-normalised texture domain in which a conventional example-based super-resolution method is used to enhance the resolution of the facial texture. The second approach is an alternative for this framework which uses the 3D information to formulate super-resolution of the facial texture in the normalised texture domain without the need for texture extraction from the LR input image. Experimental evaluation confirms that both methods are able to inject relevant HR information into the facial texture which is beneficial for recognition.

Finally, Chapter 7 concludes the thesis and suggests directions for future work.
1.4 Contributions

To the best of my knowledge, this work is the first attempt in using a 3DMM for LR face recognition. More specifically, the work presented in this thesis has resulted in the following contributions to the field of face recognition:

- An extensive survey and critical analysis of the current approaches to low-resolution face recognition.
- A novel method for fitting a high-resolution 3D Morphable Model to a low-resolution 2D image.
- An experimental evaluation of the proposed fitting approach which shows that our proposed method is capable of extracting HR information from an LR image, in the form of a high-resolution 3D model.
- An experimental evaluation of the application of 3DMM to the problem of low-resolution face recognition using the model parameters extracted from the LR input.
- A novel LDA-based approach to using 3DMM parameters in face recognition and it’s experimental validation in a face verification scenario.
- Two novel alternative approaches for 3D-assisted facial texture super-resolution in the presence of pose and illumination variations, and their application to 3D-assisted face recognition in low-resolution.

1.5 List of Publications

This work in collaboration with other researchers in the Centre for Vision, Speech and Signal Processing, at the university of Surrey, has resulted in the following publications:

Chapter 2

3D Morphable Model for Unconstrained Face Recognition

2.1 Introduction

Many research areas in various fields involve human faces. Examples include synthesising human face images in Computer Graphics and face image analysis in Computer Vision for the purpose of face recognition or pose correction. 3D morphable face models provide a unified approach addressing both analysis and synthesis of faces. Since the pioneering work of Blanz and Vetter ([105], [19], [20]), such models have been applied to various such problems.

This chapter presents a review of the 3D morphable face model and its application to unconstrained face recognition under varying poses and illuminations. We present a review of the related literature on building a 3D morphable model and the main approaches to face analysis through model fitting.

In brief, a 3D morphable face model (3DMM) is a vector space representation of 3D faces which exploits correspondences between exemplar 3D faces to learn class-specific knowledge. Using such knowledge, as well as prior knowledge of the image formation process, a generative model can be introduced which is capable of synthesising novel human face images. Also, such a model can be used to infer information from a given 2D face image. This can be achieved by estimating the model parameters such that - when used for synthesising an image - they produce a facial image similar to the input; a task known as model fitting. Model fitting is performed in an analysis by synthesis framework and the optimal model parameters obtained by model fitting can be used for tasks such as face analysis and coding.

Of particular interest to this thesis is the application of the 3D morphable face model (3DMM) to unconstrained face recognition under varying poses and illuminations. As was discussed in Chapter 1, many intrinsic and extrinsic sources of variation exist in a facial image such as facial shape, facial texture, pose, illumination, and expression. A 3DMM models most of these sources explicitly, separately, and independently; providing the opportunity for the intrinsic information to be used for face recognition.
Chapter 2. 3D Morphable Model for Unconstrained Face Recognition

Such a model can be constructed from a collection of 3D face scans each of which is defined in terms of a shape vector $S_k$, and a texture vector $T_k$, such that any scan within the collection, or any new face, can be synthesized as a linear combination of the 3D scans:

$$S_{mod} = \sum_k a_k S_k, \quad T_{mod} = \sum_k b_k T_k, \quad \sum_k a_k = \sum_k b_k = 1. \quad (2.1)$$

In order to perform a linear combination such as the one in Equation 2.1, all the shape and texture vectors must be registered such that they are in dense point-to-point correspondence. This means that a given vertex must correspond to the same location on all face scans. Blanz et al. employed a modified optical flow algorithm to put their 3D scans in dense correspondence [20]. However, their method is suited for textured 3D meshes with high polygonal density and parametrised in cylindrical coordinates (see [86]). In [86], Tena proposed an alternative approach named the Inverse Multi-Resolution Dense Registration (IMDR) method which was used for registering 3D facial scans. This method is applicable to any kind of textured 3D data.

The 3D morphable model used in this thesis was not developed as part of this work. We used the 3DMM developed in the Centre for Vision, Speech, and Signal Processing (CVSSP), University of Surrey, by Tena [86]. Throughout this thesis, when necessary to distinguish this particular morphable face model we refer to it as the CVSSP-3DMM. In the remaining parts of this chapter, a short description of the construction procedure for this model is given in Section 2.2 followed by a description of the image synthesis process using a 3DDM in Section 2.3. We then review the main model fitting methods in Section 2.4 and review a procedure for extracting texture from the original input using the fitted 3DMM in Section 2.5. Section 2.6 presents the approaches used in the literature for using a 3DMM in face recognition. Finally, in Section 2.7 we discuss our conclusions from this chapter.

2.2 Building a 3D Morphable Model

One way of constructing a model that is able to generate any individual face is to construct a face space where each point in the space represents a face. Generating a new face can then be done by sampling from this space.

Such a space can be constructed from a set of sample points, in other words, a set of exemplar faces. Therefore, the first step in constructing the 3D model is to collect a set of sample 3D face scans.

The robustness of a morphable model depends on the variability contained in its database of exemplar faces. Accordingly, the ideal database would contain an equal number of samples of male and female faces from the main ethnic groups. All samples should be collected without hair occlusions, makeup or facial expressions, since these

\footnote{This model was initially built as part of a PhD thesis ([86]), and later enhanced by adding more sample faces and increasing the resolution of the model.}
are not intrinsic to the shape or texture of faces. Furthermore, the texture should be captured under uniform illumination conditions to avoid the appearance of shadows and specularities. Collecting a database of these characteristics is a daunting and time consuming task, for which the resources were not available. Instead, the database used for constructing the CVSSP-3DMM contains 3D scans from different sources obtained for various different reasons. This database includes 168 human faces of males and females of different age groups and ethnicity including 18 face scans provided by the University of Notre Dame as a supplement to the FRGC database [82], 51 collected at CVSSP with a 3dMDFace™ System, and 99 from the SUNY database [109]. The 3D scans used to form this database do not necessarily have the above-mentioned ideal characteristics. This database is biased towards white males and was captured under non-uniform illumination conditions. The adverse effects of non-uniform illumination have been mitigated using the albedo extraction algorithm developed in [115]; however, some illumination artifacts still remain (see Figure 2.1).

2.2.1 Registered 3D Faces

As mentioned earlier, a crucial step in building a 3DMM is establishing dense correspondences across the database of face scans. Dense correspondence implies that all vectors contain the same number of 3D vertices, connected in the same way, ordered in the same manner, and located at corresponding structures of the 3D faces. Prior to this step, each raw face scan comprises a 3D mesh and a 2D RGB texture map. The \((x, y, z)\) coordinates of the vertices of the 3D mesh of the \(k^{th}\) face scan are concatenated into a shape vector \(S_{raw}^k = [x_1, y_1, z_1, x_2, y_2, z_2, ..., x_n, y_n, z_n]^T\). Similarly, a texture vector \(T_{raw}^k = [r_1, g_1, b_1, r_2, g_2, b_2, ..., r_n, g_n, b_n]^T\) is created by sampling the RGB texture map at the locations dictated by the vertices of the 3D mesh. However, for each raw face scan the value of \(n\) will be different, impeding the addition of different shape and texture vectors as suggested by equation 2.1. Moreover, the raw face scans are not aligned since they were captured with variations in pose. These problems are solved by densely registering the database using the Inverse Multi-Resolution Dense Registration (IMDR) method proposed in [86, 99]. This method, which is an enhancement of the earlier method proposed by Mao et al. [72], deforms a generic face model onto a target face to establish dense correspondence between the deformed generic faces rather than between raw 3D scans. The method comprises three stages: 1) global mapping, 2) local matching, and 3) energy minimisation. The global mapping stage makes use of a set of manually defined landmarks. By bringing these landmarks into alignment using the thin-plate spline [22] technique, this first stage establishes coarse correspondence between the two faces. In the second stage, for each vertex of the generic face, the most similar vertex of the target face within a given search radius is identified by defining a similarity measure between the vertices. Using these corresponding vertices on the two faces, an energy function is then defined and minimised in the third stage using the conjugate gradient method. The energy function is comprised of an external energy term which attracts the vertices of the generic face to their corresponding vertices on the target face; and an internal term which constrains the deformation of the generic face, thus maintaining the original local mesh structure. The enhancements made by Tena et al. to the original algorithm include modifications to the similarity measure used for
finding the most similar vertex and using an adaptive search radius during the second stage, as well as using an iterative coarse-to-fine strategy during the second and third stages while enforcing symmetry at the coarse scale. The enthusiastic reader is referred to [86] and [99] for more details about the IMDR algorithm.

After registration and alignment a new set of shape and texture vectors can be defined for the \( k \)th registered face scan:

\[
S_k = [s_1^T, s_2^T, \ldots, s_{N_v}^T]^T = [x_1, y_1, z_1, x_2, \ldots, y_{N_v}, z_{N_v}]^T,
\]

\[
T_k = [t_1^T, t_2^T, \ldots, t_{N_v}^T]^T = [r_1, g_1, b_1, r_2, \ldots, g_{N_v}, b_{N_v}]^T,
\]

(2.2)

where \( s_i = [x_i, y_i, z_i]^T \) (with \( i = 1, 2, \ldots, N_v \)) is a vector of 3D Cartesian coordinates of the \( i \)th vertex, \( t_i = [r_i, g_i, b_i]^T \) is a vector of its RGB texture values, and \( N_v \) is the number of vertices of the generic model used for registration. Using a generic model with \( N_v \) vertices, the registered faces, and therefore the 3DMM, will also have \( N_v \) vertices. In our case, the generic model has more than 29,000 vertices. Figure 2.1 shows two samples of the 3D database before and after dense registration and alignment. Since each face is represented by vectors of a set of discrete vertices, the face space will be a vector space. Given the shape (\( S_k \)) and texture (\( T_k \)) vectors, any new face can then be constructed by linear combination of the existing vectorised faces.

The vertices of each face are connected in a triangular mesh with the same triangle list for all faces. This list stores for each triangle of the mesh the three vertices that constitute it. Thus, after registration, all faces have the same number of vertices (\( N_v \)) connected in the same manner (through the triangle list) and each vertex corresponds to the same facial location on all faces.

### 2.2.2 PCA-Based Face Space

In order to use the face vector space appropriately, an orthogonal basis and a probability distribution over the face space is required. Principal Components Analysis (PCA) is a statistical tool that satisfies both of these requirements. Using PCA the data is projected to a new space with orthogonal axes in which the covariance matrix of the projected data is diagonal, i.e. the data is decorrelated. Furthermore, the probability distribution of the data is readily available and can be expressed in terms of the coefficients of the projected data.

PCA is generally used as a dimensionality reduction tool to project data from an \( N \)-dimensional space to a \( D \)-dimensional space where \( D < N \). It can be derived from two perspectives [14]. The first formulation is the maximum variance formulation, which derives the transform with the aim of projecting the data to a lower dimensional space in which the variance of the data is maximised. It can be shown that the axes of the new space are ordered by the amount of projected variance, so a subspace formed of the first \( D \) axes is the subspace containing the highest amount of variance which can be preserved in \( D \) dimensions. The second formulation derives the transform by searching for the lower dimensional space in which the reconstruction error is minimised. With
2.2. Building a 3D Morphable Model

Figure 2.1: CVSSP database exemplars before and after registration. The raw face scans as captured from the sensor, the registered shape mesh, and the registered albedo texture map are shown.
either formulation, the same transform is derived in which the transformation matrix is formed by the eigenvectors of the covariance matrix of the training data.

Assuming shape and texture are independent, each is analysed separately. Shape alone forms a vector space, and texture alone forms another vector space. The face space is then the combination of these two spaces. PCA is applied separately to shape and texture. We describe the application of PCA to the shape data; its application to texture is straightforward.

Subtracting the average shape \( \bar{S} \) from each training vector and organising the training data in a data matrix, \( A \), we have:

\[
\bar{S} = \frac{1}{M} \sum_{k=1}^{M} S_k, \\
a_k = S_k - \bar{S}, \\
A = [a_1, a_2, \ldots, a_M]
\] (2.3)

The covariance matrix of the training data is then given as \( C_A = \frac{1}{M} AA^T \) and the transformation matrix, \( U_s \), can be obtained by solving the eigen decomposition equation²:

\[
C_A U_s = U_s \Sigma_s
\] (2.4)

where columns of the orthogonal matrix \( U_s \) are normalised eigenvectors of the covariance matrix and \( \Sigma_s \) is a diagonal matrix with the eigenvalues on the main diagonal.

Now, instead of representing the data matrix \( A \) in its original space, it can be projected to the space spanned by the columns of \( U_s \). Let us denote by the matrix \( B \) this new representation of the data, where:

\[
B = U_s^T A
\] (2.5)

Given that \( U_s \) is orthonormal, it is straightforward to show that the covariance matrix of the projected data, \( C_B \), is diagonal:

\[
C_B = \frac{1}{M} BB^T = \Sigma_s
\] (2.6)

In other words, the projected data are decorrelated. Furthermore, it is apparent from Equation 2.6 that the variance projected to the \( i^{th} \) PCA axis is equal to the \( i^{th} \) eigenvalue of the covariance matrix of the original data. Hereafter, we denote by \( \sigma^2_S \), and \( \sigma^2_T \) the projected variances of, respectively, the shape and the texture vectors.

Furthermore, given the orthonormality of \( U_s \), the original data can be reconstructed by projecting the data from the PCA space back to original space:

²In practice the eigenvectors of \( C_A \) are found much more efficiently by first finding the eigenvectors of \( C_A^T = A^T A \). See Section 12.1.4 of [14] for details.
2.2. Building a 3D Morphable Model

\[ \mathbf{A} = \mathbf{U}_s \mathbf{B} \]  

(2.7)

In the above, it is assumed that all the dimensions are being used and PCA is only used to map the data into a new coordinate system in which the data is decorrelated. However, as mentioned earlier, PCA can also be used to reduce the data dimensionality. In our case, the shape data used to train the PCA transform is the set of shape vectors \( \mathbf{S}_k = [x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_{N_v}, y_{N_v}, z_{N_v}]^T \) where \( k = 1, 2, \ldots, M \). The dimensionality of each training vector is \( 3N_v \). However, the number of training samples (\( M \)) is far less than the dimensionality of the data points which means that our training data lies in a linear sub-space of the input \( 3N_v \)-dimensional space and that the dimensionality of this linear sub-space is at most \( M - 1 \). That is, at most \( M - 1 \) eigenvalues are non-zero.

PCA can be used to reduce the data dimensionality to \( D_s < M - 1 \). To this end, the transformation matrix is formed of the first \( D_s \) eigen vectors of the covariance matrix (ordered by the size of the corresponding eigenvalues). In this case, the reconstruction represented by Equation 2.7 would result in an estimate of \( \mathbf{A} \) and some reconstruction error would occur. It can be shown ([14]) that the subspace spanned by the first \( D_s \) columns of \( \mathbf{U}_s \), is the best \( D_s \)-dimensional subspace to represent the data in terms of minimum reconstruction error. In other words, by keeping only the first \( D_s \) dimensions of the new space, one can reduce the dimensionality of the data to \( D_s \), while keeping the mean squared error of the reconstructed data points at minimum.

In order to reduce the dimensionality of the shape and texture vectors, a separate PCA is applied to each, with the condition that the reconstruction error is limited to 1%. In the particular dataset used for training the CVSSP-3DMM, this yields \( D_s = 55 \) and \( D_T = 123 \) as the dimensionality of, respectively, the shape and texture vector spaces. Finally, the morphable model’s shape and texture can be represented as:

\[ \mathbf{S} = \bar{\mathbf{S}} + \sum_{i=1}^{D_s} \alpha_i \mathbf{S}_i^e, \quad \mathbf{T} = \bar{\mathbf{T}} + \sum_{i=1}^{D_T} \beta_i \mathbf{T}_i^e, \]  

(2.8)

where \( \mathbf{S}_i^e \) and \( \mathbf{T}_i^e \) are the \( i^{th} \) shape and texture eigenvectors, respectively. We denote by \( \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_{D_s}]^T \) and \( \beta = [\beta_1, \beta_2, \ldots, \beta_{D_T}]^T \) the vectors of shape and texture coefficients, respectively. We refer to these coefficients as the model’s shape (\( \mathbf{S} \)) and texture (\( \mathbf{T} \)) parameter vectors. Figure 2.2 illustrates the average shape and texture of the 3DMM along with shape and texture variations caused by changing the first three \( \alpha \) and \( \beta \) parameters respectively.

The probability densities of parameters \( \alpha \) and \( \beta \), or equivalently, the densities for the model’s shape and texture, are given by:

\[ p(\mathbf{S}) = p(\alpha) \sim \exp \left[ -\frac{1}{2} \sum_{i=1}^{D_s} \frac{\alpha_i^2}{\sigma_{S,i}^2} \right], \quad p(\mathbf{T}) = p(\beta) \sim \exp \left[ -\frac{1}{2} \sum_{i=1}^{D_T} \frac{\beta_i^2}{\sigma_{T,i}^2} \right], \]  

(2.9)

where \( \sigma_{S,i}^2 \) and \( \sigma_{T,i}^2 \) are the variances of the \( i^{th} \) shape component, and the \( i^{th} \) texture component, respectively.
Figure 2.2: Average and first three principal components of the 3DMM, visualised by adding $\pm 2\sigma_S$ and $\pm 2\sigma_T$ to the average model.
2.2.3 Segmented Model

Any subspace learning method such as the one described above to learn the face space is limited by the amount of available training data used for its training. The model used in this thesis (CVSSP-3DMM) is derived from statistics learnt using 168 face scans, and as mentioned earlier the dimensionalities of the shape and texture spaces are $D_s = 55$ and $D_T = 123$, respectively. This may not be enough to account for the variations present in human faces. Therefore, the expressiveness of a model built using the above procedure would be limited. One way to overcome such limits is to increase the number of training scans. However, only a limited number of scans was available for building the model. Blanz and Vetter [20] proposed to segment the model into 4 regions and use separate shape and texture parameters to encode each segment. A similar approach is used in this work where we segment the face into four regions: eyes, nose, mouth, and the rest of the face. Figure 2.3 shows these regions. Note that we do not build a separate PCA model for each segment. Instead, we use the same global model to analyse each of the segments separately. More specifically, we fit the model to each segment separately by evaluating the fitting cost function only over the region corresponding to each segment\(^3\).

![Figure 2.3: The four segments of the segmented model correspond to the eyes, nose, mouth, and the rest of the face.](image)

Throughout this thesis we denote the model’s shape and texture parameter vectors by $\alpha$ and $\beta$, respectively, when they can be used interchangeably for the global and the segmented parts of the face. When necessary to distinguish them, we use $\alpha^g$ for the global model (whole face) and $\alpha^{s1}, \ldots, \alpha^{s4}$ for the four segmented parts (similar notation will be used for texture parameters).

2.3 3DMM for Image Synthesis

The model’s shape and texture values are given by the parameters $\alpha$ and $\beta$. Given these parameters and a set of rendering parameters, the 3D model can be used to render a 2D image of the face. This process involves projecting the vertices of the model to the image plane and determining the colour value at the respective image points.

\(^3\)See Section 2.4 for details of the fitting algorithm.
2.3.1 Shape projection

The first step to render an image of the 3DMM is to find the 2D image locations of its vertices. The 3D locations of the model’s vertices are determined by $\alpha$ in object-centred coordinates. A rigid transformation (3D rotation and 3D translation) maps the object-centred coordinates of the $i^{th}$ vertex, $s_i = [x_i, y_i, z_i]^T$, to a 3D coordinate frame relative to the camera:

$$w_i = R_{\theta_z}R_{\theta_y}R_{\theta_x}s_i + t_w$$ (2.10)

We refer to this camera-centred coordinates as the world coordinates. In Equation 2.10, the angles $\theta_x$, $\theta_y$, and $\theta_z$ define rotations around the $X$, $Y$, and $Z$ axes, and $t_w$ is a 3D translation. The world coordinates of vertex $i$, $w_i = [w_{x,i}, w_{y,i}, w_{z,i}]^T$, are then mapped to the the 2D image plane coordinates $p_i = [p_i, q_i]^T$ using a perspective projection:

$$p_i = p_o + f \frac{w_{x,i}}{w_{z,i}}, \quad q_i = q_o - f \frac{w_{y,i}}{w_{z,i}}$$ (2.11)

where $f$ is the focal length of a virtual camera located at the origin, pointing in the $-Z$ direction, and $p_o = [p_o, q_o]^T$ denotes the position of the optical axis on the image plane.

The shape projection described above maps a vertex from the model’s object-centred coordinates to the image frame. We denote this mapping by the vector-valued function $p(x, y, z; \tau)$, where $[x, y, z]^T$ is a vector of the 3D coordinates of the given vertex, and $\tau = [\theta_x, \theta_y, \theta_z, t_{w,x}, t_{w,y}, t_{w,z}, f]^T$ is a vector containing all the necessary parameters for the 3D rigid transform and the perspective projection described above. We refer to these parameters as the projection parameters. This mapping is continuous in $\tau$, but discrete in $(x, y, z)$. Continuity in the $(x, y, z)$ space can be achieved by using the triangle list and interpolating between the vertices.

2.3.2 Inverse Shape Projection

In order to render an image, we need the inverse of the mapping described above; that is, for each pixel of the image its 2D image plane locations should be mapped backwards to the corresponding location on the 3D model’s surface (object-centred coordinates).

It is not easy to define such an inverse mapping analytically, but it can be computed using the triangle list. By definition, the inverse mapping, $p^{-1}(p, q; \tau)$, maps a 2D image point, $[p, q]^T$, to the corresponding 3D location on the model’s surface. This inverse mapping is defined such that under the same set of parameters, the shape projection composed with its inverse is equal to the identity.

We denote the composition of a shape projection and its inverse by the symbol $\circ$. That is, $p(x, y, z; \tau) \circ p^{-1}(p, q; \tau)$ is the same as $p(p^{-1}(p, q; \tau); \tau)$, but we prefer the former notation for clarity. Thus, the aforementioned definition for the inverse mapping can be expressed as:
\[
\begin{align*}
\mathbf{p}(x, y, z; \tau) &\circ \mathbf{p}^{-1}(p, q; \tau) = [p, q]^T \\
\mathbf{p}^{-1}(p, q; \tau) &\circ \mathbf{p}(x, y, z; \tau) = [x, y, z]^T
\end{align*}
\tag{2.12}
\]

Given the above conventions, the inverse shape projection can be computed using the model’s triangle list. A point \( \mathbf{p} = [p, q]^T \) on the image plane lies inside a single triangle. The corresponding 3D point under the projection \( \mathbf{p}^{-1}(p, q; \tau) \) should lie in the same relative position to the same triangle on the model’s surface. Therefore, by determining the triangle in which a point lies on the image plane, one can express the image location of the point relative to the triangle’s vertices (barycentric coordinates) and work out the point’s location on the model’s surface using the coefficients of this barycentric representation and the 3D object-centred coordinates of the triangle’s vertices.

Figure 2.4: Forward shape projection, \( \mathbf{p}(x, y, z; \tau) \) (illustrated with green solid arrow), takes each vertex from the object-centred coordinate frame \( (x_i, y_i, z_i) \) to the image frame coordinates \( (p_i, q_i) \). Using the mapping of the three corners of a triangle, the inverse mapping can be obtained for points within the triangle. The inverse mapping, \( \mathbf{p}^{-1}(p, q; \tau) \) (illustrated with dashed red arrow), maps a given point from the image frame coordinates to the object-centred coordinates.
2.3.3 Illumination and Colour Transformation

In order to render an image from the model, one needs to determine the observed colour at each pixel’s centre. We have already established a procedure to project each pixel centre back to the model’s surface, so now we need to determine the observed colour at the obtained model point. This can be determined using the surface albedo at the given point, the local surface shape, and the scene illumination. We describe the illumination calculations at the location of a model vertex, but it can be extended to any arbitrary point on the model’s surface.

Assuming that the model was constructed using pure albedo of the faces (by capturing faces under uniform illumination or by de-illuminating the texture of the captured faces), the colour values of each point are computed using Phong’s reflection model [42] which yields the observed colour value by modeling the ambient, diffuse, and specular reflections on a surface. An ambient and a directional light are used to illuminate the scene. The ambient light is characterised by an intensity \( (L^a_r, L^a_g, L^a_b) \) that is identical at every point in the scene. The diffuse component simulates light that is coming from a particular direction and is reflected equally in all directions after striking a surface. This kind of reflection occurs on rough surfaces, which are sometimes called Lambertian surfaces, since they can be mathematically modeled by Lambert’s law [6]. The specular component simulates light which is directed as well, but is reflected in only one direction. The specular component creates the highlights that are observed on illuminated smooth shiny surfaces. The fraction of the incoming specular light that is reflected is defined by the reflection coefficient \( k_s \) \((0 < k_s < 1)\); and the surface shininess is represented by the coefficient \( \nu \), which determines the broadness of the reflected highlights [6]. The diffuse and specular light intensities are assumed to be equal and both equal to the intensity of the directional light: \( (L^d_r, L^d_g, L^d_b) \) . For simplicity, the parameters \( k_s \) and \( \nu \) are fixed to constant values.

We denote by vector \( t_i \) the RGB albedo of vertex \( i \), which is determined given the model’s texture parameters \( \beta \) (Equation 2.8). Using Phong’s model, the colour of this vertex after being illuminated by the ambient and directional lights is given as:

\[
t_i^I = \begin{pmatrix} L^a_r & 0 & 0 \\ 0 & L^a_g & 0 \\ 0 & 0 & L^a_b \end{pmatrix} \cdot t_i + \begin{pmatrix} L^d_r & 0 & 0 \\ 0 & L^d_g & 0 \\ 0 & 0 & L^d_b \end{pmatrix} \cdot (\langle n_i^{v,w}, d \rangle \cdot t_i + k_s \cdot \langle r_i, v_i \rangle ^\nu \cdot 1_{3x1})
\] (2.13)

where \( \langle ., . \rangle \) is a dot product. The first term of this equation is the contribution of the ambient light. The first term of the parenthesis simulates the diffuse component of the directional light and the last term is its specular reflection.

Figure 2.5 illustrates the four unit vectors used in Equation 2.13. Here, \( d \) is a unit vector pointing in the direction of the directional light source. The directional light source is assumed to be located at infinity and its direction is given by two angles \( \theta_l \) and \( \phi_l \) defining the latitude and longitude on a sphere centred at the origin. Accordingly, the vector \( d \) is given in Cartesian coordinates as:
2.3. 3DMM for Image Synthesis

Figure 2.5: Unit vectors used in the Phong model: surface normal ($n$), light direction ($d$), viewing direction ($v$), and reflection direction ($r$).

\[
d = \begin{pmatrix}
\cos \theta \sin \phi \\
\sin \theta \\
\cos \theta \cos \phi
\end{pmatrix}
\]  

(2.14)

Also, in Equation 2.13, $n_{v,w}^i$ is the normal of the vertex $i$, expressed in world coordinates; hence, the superscript $w$. Furthermore, the superscript $v$ is to highlight that this is a normal vector of a vertex as opposed to a normal vector of a triangle which we will denote by superscript $t$. The vector $n_{v,w}^i$ can be obtained by rotating the normal from the object-centred coordinates (denoted by $n_{v,o}^i$) to the world coordinates:

\[
n_{v,w}^i = R_{\theta_z}R_{\theta_y}R_{\theta_x}n_{v,o}^i
\]  

(2.15)

The normal vector of a vertex in object-centred coordinates, $n_{v,o}^i$, can be defined as the average of the normals of the triangles connected to it, normalised to unit length:

\[
n_{v,o}^i = \frac{\sum_{j \in T_i} n_{t,o}^j}{\|\sum_{j \in T_i} n_{t,o}^j\|}
\]  

(2.16)

where $T_i$ is the set of triangles for which vertex $i$ is one of the three corners. Alternatively, one can use a weighted average where the contribution of the normal of each triangle is weighted by its area. Denoting the object-centred coordinates of the three vertices that form the $j^{th}$ triangle by $s_{j,1}$, $s_{j,2}$, and $s_{j,3}$, the normal of this triangle is given by the cross-product of two of its edges, normalised to unit length:
\[ n_{j}^{t, o} = \frac{(s_{j,1} - s_{j,2}) \times (s_{j,1} - s_{j,3})}{\| (s_{j,1} - s_{j,2}) \times (s_{j,1} - s_{j,3}) \|} \]  

(2.17)

The vector \( v_i \) in Equation 2.13 is a unit vector in the direction from vertex \( i \) to the viewer (the camera’s centre). Here, the camera’s centre is assumed to be located at the origin of the world coordinate frame; therefore, the vector \( v_i \) is given by:

\[ v_i = -\frac{w_i}{\| w_i \|} \]  

(2.18)

where \( w_i \) is the world coordinates of vertex \( i \) (Equation 2.10).

Finally, the unit vector \( r_i \) in Equation 2.13 points in the direction that a perfectly reflected light beam coming from direction \( d \) would take. Given the normal \( (n_{i}^{v,w}) \) and the light direction \( (d) \) vectors, the reflection direction \( (r_i) \) can be computed as:

\[ r_i = 2 \langle n_{i}^{v,w}, d \rangle n_{i}^{v,w} - d \]  

(2.19)

It is worth mentioning at this point that in the above we merely described the process of computing the colour value of a vertex of the model when illuminated by ambient and directional light. However, the illuminated texture of model points which lie on the flat area within a certain triangle can be determined in a similar fashion, with only minor differences. More specifically, if the point at hand lies inside triangle \( j \), the normal vector at the given point is equal to the triangle’s normal. Therefore, instead of \( n_{i}^{v,w} \), the corresponding triangle’s normal, \( n_{j}^{t,w} \), will be used in Equation 2.13. Also, instead of \( t_i \) which defines the albedo at a certain vertex, the albedo of the given point must be used which can be computed by interpolating between the albedos of the three corners of the corresponding triangle.

**Colour transformation:** Images captured with different cameras have variations in the tone of colour. In order to handle a variety of colour images as well as grey scale images, Blanz and Vetter [19] proposed to use a colour transformation which applies channel-specific gains \((g_r, g_g, g_b)\) and offsets \((o_r, o_g, o_b)\) as well as a colour contrast, \(c\), to the illuminated texture. Hence, the final colour value of each vertex, \( t_i^C \), is determined using the following linear transform:

\[ t_i^C = G t_i^I + o, \]  

(2.20)

where:

\[ G = \begin{pmatrix} g_r & 0 & 0 \\ 0 & g_g & 0 \\ 0 & 0 & g_b \end{pmatrix}, \quad \left[I + (1 - c) \begin{pmatrix} 0.3 & 0.59 & 0.11 \\ 0.3 & 0.59 & 0.11 \\ 0.3 & 0.59 & 0.11 \end{pmatrix} \right], \quad \text{and} \quad o = [o_r, o_g, o_b]^T \]  

(2.21)

The above procedure describes how to compute the colour value for a given point on the model’s surface. We denote this procedure as a vector-valued function \( t^C(x, y, z; \beta, \gamma, \alpha, \tau) \)
where \( \gamma = [L^a_r, L^a_g, L^d_r, L^d_g, \theta_l, \phi_l, g_r, g_g, g_b, c, o_r, o_g, o_b]^T \) is a vector containing all the illumination and colour transformation parameters. From this point on we will refer to all these parameters as the illumination parameters. Similar to the shape projection function, this function is also discrete in the \((x, y, z)\) space and continuity can be provided by interpolation using the triangle list. Note that the illuminated and colour corrected texture depends, not only on the model’s texture parameters (\(\beta\)) and the illumination parameters (\(\gamma\)), but also on the shape (\(\alpha\)) and projection (\(\tau\)) parameters which are used to find the normal and viewing direction of the points.

**Occlusion and shadow**: We described the process of projecting model vertices to the image plane and determining their illuminated colour. Given a particular viewing direction, only a subset of the model vertices will be visible from the camera viewpoint, the rest will be hidden due to self-occlusion. The set of visible vertices can be determined using a z-buffer algorithm [26] which is an algorithm commonly used in Computer Graphics for handling occlusions.

Furthermore, we have so far assumed that all points receive the same amount of light from the directional light source. However, depending on the relative position of the light source and the model in 3D space, some parts of the model’s surface will be in shadow. There are two scenarios which result in a point being in shadow and these are modelled in quite different ways. The two types of shadows resulting from these scenarios are known as attached shadows and cast shadows. An attached shadow (also called a self-shadow) occurs when a point on the surface is oriented away from the light source, thereby occluding itself from the illumination. Shadows of this sort are easily modeled, since they depend only upon the local geometry of the surface (the normal direction). In order to determine whether a point is in attached shadow, it is sufficient to find the normal direction at the point and, if the normal is pointing away from the light, the point will be in shadow. To take account of the attached shadows, the two scalar products of Equation 2.13, \( \langle \mathbf{n}_i, \mathbf{d} \rangle \) and \( \langle \mathbf{r}_i, \mathbf{v}_i \rangle \), are lower bound to zero. On the other hand, cast shadows are caused when an entirely different region of the surface intersects the path from the light source to the point at hand. Cast shadows are more complex to model since they are dependent on the global geometry of the surface. In order to find the points which are subject to cast shadow, we render the model from the light view-point and use a z-buffer algorithm to find whether the given point is occluded by other parts of the model from the light view-point.

If a point is in shadow, it will only be illuminated by the ambient light. Therefore, in Equation 2.13, only the first term will be used to determine the illuminated colour value (\( t^I \)) for such a point.

### 2.3.4 Image synthesis

We now have all the necessary tools to render an image of the model. Synthesising an image, \( I_{\text{model}}(p_j, q_j; \beta, \gamma, \alpha, \tau) \), is performed by mapping the illuminated and colour transformed texture from the model space to the image plane using an inverse shape projection, as:
\[ I_{\text{model}}(p_j, q_j; \beta, \gamma, \alpha, \tau) = t^C(x, y, z; \beta, \gamma, \alpha, \tau) \circ p^{-1}(p_j, q_j; \tau) \] (2.22)

where \([p_j, q_j]^T\) is the 2D image coordinates of the centre of the \(j^{th}\) pixels while the index \(j\) runs over the pixels for which an inverse shape projection exists. In other words, the pixels for which \(p^{-1}(p_j, q_j; \tau)\) is a visible point of the model.

### 2.4 Model Fitting to a 2D Image

As was mentioned previously, many different sources of variation exist in a face image. These include information such as facial shape and texture which are intrinsic to the identity as well as information such as scene illumination, camera view point (pose) etc. In Section 2.3, we described the process of synthesising a facial image using a 3DMM given all the above information. The aim of model fitting is to reverse this process. In other words, given a 2D face image, we want to infer various types of intrinsic and extrinsic information. Ideally, the fitting algorithm should be able to separate the effects of various types of information which are convolved in the 2D image. For instance, given the illuminated colour, it should be able to separate the effects of illumination and facial texture.

More specifically, considering the image synthesis model of Equation 2.22, the aim is to infer the model shape and texture parameters (\(\alpha\) and \(\beta\)) as well as the set of all rendering parameters (\(\tau\) and \(\gamma\)) such that an image synthesised using these parameters will appear similar to the input image. For brevity let us denote the set of all model parameters by \(\mu = \{\alpha, \beta\}\) and the set of all rendering parameters by \(\rho = \{\tau, \gamma\}\).

#### 2.4.1 Stochastic Newton Optimisation

The first algorithm to address this problem was the Stochastic Newton Optimisation (SNO) proposed by Blanz and Vetter [19]. They used an analysis-by-synthesis approach in which the model is used to render an image in the synthesis step and the error between the rendered and the input image is used in the analysis step to correct the sought parameters. The pixel colours of the rendered image are supposed to be as close as possible to the input image in terms of Euclidean distance. Therefore, the aim is to minimise the following error (cost function) which depends on the pixel colours:

\[
E_c = \sum_{p,q} \|I^{\text{input}}(p, q) - I^{\text{model}}(p, q)\|^2 \] (2.23)

In Equation 2.23, \(I^{\text{model}}(p, q)\) is the value of the rendered image at pixel location \((p, q)\) which is given by equation 2.22. Hence, the cost function of Equation 2.23 can be written as:

\[
E_c = \sum_{p,q} \|I^{\text{input}}(p, q) - t^C(x, y, z; \mu, \rho) \circ p^{-1}(p, q; \tau)\|^2 \] (2.24)
However, as was mentioned earlier, it is not feasible to express the inverse shape mapping analytically. Composing the terms inside the norm in Equation 2.24 from the right by $p(x, y, z; \tau)$ and sampling points on the model's surface instead of the image yields an equivalent cost function which depends on the forward shape mapping:

$$E_c = \sum_i \|I_{input}(p, q) \circ p(x_i, y_i, z_i; \tau) - t^C(x_i, y_i, z_i; \mu, \rho)\|^2$$  \hspace{1cm} (2.25)$$

where $\{[x_i, y_i, z_i]^T | i=1,2,\ldots \}$ is a set of points sampled on the model's surface. Sampling the model at the vertices is not the best choice as computing the unit-length normal and its derivative with respect to shape parameters at a vertex is computationally expensive. A better choice would be to sample the model at the centroids of the polygons. Hence, a shape and texture model expressed at the triangle centres is created using interpolation between the shape and texture principal components of the three corners of the triangle. Then, the index $i$ in Equation 2.25 runs over the visible triangles of the model, and $[x_i, y_i, z_i]^T$ is the 3D object-centred coordinates of the centre of the $i^{th}$ triangle.

According to Equation 2.25, in order to calculate the residual pixel colour error, the illuminated and colour-transformed colour at the centre of each triangle is compared with the corresponding colour value of the input image at the location obtained by projecting the triangle’s centre to the image plane.

Fitting the model to a 2D image is an ill-posed problem. It is therefore essential to constrain the set of solutions. This can be done through a Maximum Aposteriori (MAP) estimation framework. The fitting problem can be viewed as maximising the posterior distribution of the sought parameters given the input image:

$$\{\mu^*, \rho^*\} = \operatorname{argmax}_{\mu, \rho} \{p(\mu, \rho | I_{input})\} = \operatorname{argmax}_{\mu, \rho} \{p(I_{input} | \mu, \rho)p(\mu)p(\rho)\}$$  \hspace{1cm} (2.26)$$

In Equation 2.26, it has been assumed that the model parameters, $\mu$, and the rendering parameters, $\rho$, are statistically independent, which is a plausible assumption.

For independent, identically distributed ($i.i.d$) Gaussian pixel noise with a standard deviation $\sigma_c$, the likelihood of observing the input image, $I_{input}$, given the sought parameters can be expressed as $p(I_{input} | \mu, \rho) \sim \exp\{\frac{-1}{2\sigma_c^2} E_c\}$. Also, assuming statistical independence between the shape and texture parameters, the prior distribution of the model parameters can be expressed as $p(\mu) = p(\alpha)p(\beta)$, where the prior distributions for $\alpha$ and $\beta$ are given in Equation 2.9. Finally, $p(\rho)$ is also assumed to be a normal distribution with ad hoc values for the mean, $\bar{\rho}_j$, and standard deviation, $\sigma_{\rho,j}$, of the $j^{th}$ rendering parameter. The posterior distribution is then maximised by minimising the following cost function:

$$E_{SNO_1} = \frac{1}{\sigma_c^2} E_c + \sum_{j=1}^{D_z} \frac{\alpha_j^2}{\sigma_j^2} + \sum_{j=1}^{D_T} \frac{\beta_j^2}{\sigma_{T,j}^2} + \sum_j \frac{(\rho_j - \bar{\rho}_j)}{\sigma_{\rho,j}^2}$$  \hspace{1cm} (2.27)$$
A gradient descent method is used to optimise the above criterion which uses analytical derivatives of the cost function and updates the parameters as \( \alpha_j \rightarrow \alpha_j - \lambda_j \frac{\delta E_{SNO_j}}{\delta \alpha_j} \), with suitable values for \( \lambda_j \). The parameters \( \beta_j \) and \( \rho_j \) are also updated similarly. In this gradient descent method, the contributions of all different triangles of the model would be redundant. Therefore, in each iteration a random subset \( K \subset \{1, 2, \ldots, N_v\} \) of 40 triangles are chosen and \( E_c \) is estimated using only these triangles. This stochastic approach not only improves the computational efficiency, but also helps to avoid local minima by adding noise to the gradient estimate [19]. The probability of selecting a particular triangle is proportional to its area on the image plane. The areas of all triangles as well as the occlusions and shadows are calculated at the beginning of the optimisation and once every 1,000 iterations.

The cost function in Equation 2.27 suffers from multiple local minima. In order to avoid being trapped in local minima the algorithm follows a coarse-to-fine strategy in several respects:

- The first iterations of the algorithm are performed using a down-sampled version of the input image with a low-resolution 3DMM.
- The algorithm starts by optimising only the first few parameters \( \alpha_j \) and \( \beta_j \) along with all parameters \( \rho_j \). In subsequent iterations, more shape and texture parameters are added gradually.
- The algorithm starts with a relatively large \( \sigma_c \) which puts a strong weight on the prior probability and ties the optimum towards the prior expectation value. Later, \( \sigma_c \) is decreased to achieve maximum fitting quality.
- In the final iterations, after fitting the global model, the parameters \( \alpha_j \) and \( \beta_j \) are optimised independently for each segment, with parameters \( \rho_j \) fixed.

In [20], Blanz and Vetter enhanced the above SNO algorithm by adding an additional term to the cost function of Equation 2.27 which uses a set of manually-defined anchor points. The user selects a set of vertices corresponding to facial features and annotates their corresponding location on the input image. The optimal parameters should project these selected vertices to their 2D image location. Accordingly, the anchor point cost function is defined as the Euclidean distance between the projection of these vertices and the corresponding locations annotated by the user:

\[
E_a = \sum_j \left\| \left( \hat{p}_j - \hat{q}_j \right) \right\|^2
\]

(2.28)

where \([\hat{p}_j, \hat{q}_j]^T\) is the 2D location of the \( j^{th} \) landmark annotated by the user on the input image, and \([p_{k_j}, q_{k_j}]^T\) is the location of its corresponding vertex, \( k_j \), after projection to the image plane through forward shape projection. Assuming that the coordinates of the anchor points are subject to Gaussian noise with zero mean and standard deviation \( \sigma_a \), the final cost function to be optimised is:
2.4. Model Fitting to a 2D Image

\[ E_{\text{SNO2}} = \frac{1}{\sigma_c^2} E_c + \frac{1}{\sigma_a^2} E_a + \sum_{j=1}^{D_S} \frac{\alpha_j^2}{\sigma_S^2 j} + \sum_{j=1}^{D_T} \frac{\beta_j^2}{\sigma_T^2 j} + \sum_j \frac{(\rho_j - \bar{\rho}_j)}{\sigma_{\rho,j}^2} \]  \quad (2.29)

Ad hoc values are used for \( \sigma_a \). The initial iterations of the algorithm put a high weight on \( E_a \) and the prior terms. The final iterations put more weight on \( E_c \) and no longer rely on \( E_a \).

2.4.2 Multi-Feature Fitting

Inspired by classifier combination in the field of pattern classification, Romdhani and Vetter [87, 88] proposed a fitting algorithm that is more robust to the local minima problem. In classifier combination, multiple classifiers are combined to obtain a classifier that has better performance than any of the individual classifiers. Similarly, as argued by Romdhani and Vetter [87], a fitting algorithm can be more robust to the imaging condition or the local minima problem if it was based on multiple features. So instead of maximising \( p(\mu, \rho | I^{\text{input}}) \), as does the SNO algorithm, they proposed a framework which uses a number of features, \( f^i(I^{\text{input}}) \), extracted from the input image and maximises \( p(\mu, \rho | f^1(I^{\text{input}}), f^2(I^{\text{input}}), \ldots, f^N(I^{\text{input}})) \). This method is referred to as the Multi-Feature Fitting (MFF) framework.

The MFF algorithm uses a MAP framework to maximise the posterior distribution of the model and the rendering parameters, given not only the input image, but also a number of features extracted from it. To make the notation more readable, we derive the formulation using two features, \( f^1 \) and \( f^2 \), and make their dependence on the input image implicit. According to Bayes’ rule, the posterior distribution to be maximised can be written as:

\[ p(\mu, \rho | f^1, f^2) = \frac{p(f^1 | \mu, \rho) p(f^2 | \mu, \rho) p(\mu, \rho)}{p(f^1, f^2)} \]  \quad (2.30)

Assuming that 1) the features are independent 2) deterministic feature extractors are used, and 3) the model parameters and the rendering parameters are statistically independent, the above can be written as:

\[ p(\mu, \rho | f^1, f^2) = p(f^1 | \mu, \rho) p(f^2 | \mu, \rho) p(\mu) p(\rho) \]  \quad (2.31)

Maximising the above equation is equivalent to minimising its negative logarithm. Hence, the optimal parameters can be found as:

\[ \{\mu^*, \rho^*\} = \arg\min_{\mu, \rho} \left\{ -\ln p(f^1 | \mu, \rho) - \ln p(f^2 | \mu, \rho) - \ln p(\mu) - \ln p(\rho) \right\} \]  \quad (2.32)

In Equation 2.32, each \( -\ln (.) \) defines a cost function that depends on a particular feature. Although the formulation above was derived using only two features, it is
straightforward to extend it to arbitrary numbers of features and cost functions. Opti-
mally, each cost function should have local minima in different areas of the parameter
space compared to the others. Then, the combined cost function would have fewer local
minima and a smoother behaviour over the whole parameter space.

The cost functions used in the MFF work of Romdhani and Vetter include cost functions
depending on: pixel colours, edges, anchor points, specular highlights, prior probabili-
ties, and texture constraints. In the following, we will give a brief description for each
and we refer the reader to [87] and [88] for a more detailed discussion about each feature
and its characteristics.

**Pixel Colour** The feature used to fit the pixel colour is simply the input image itself:

\[
f_c(\text{Input}(p,q)) = \text{Input}(p,q) \tag{2.33}
\]

This is the same feature that was used in the SNO algorithm. With similar assumptions
and derivation procedure as was described previously for the SNO algorithm, the cost
function associated with the pixel colour is similar to the pixel colour cost of the SNO
algorithm \( E_c \), given in Equation 2.25.

**Edges** The edge cost function can be used to optimise the shape and projection
parameters such that the edges of the model face image correspond to the edges of the
input image. A deterministic edge detector; namely, the *canny* edge detector, is used
to find the edges of the input image. This provides a binary edge map of the input
image:

\[
f_e(\text{Input}(p,q)) = \text{canny}(\text{Input}(p,q)) \tag{2.34}
\]

Each edge pixel of the input image is assigned an index \( j = 1, \ldots, J \) and their 2D
positions are referenced by \( e^I_j \). Let us denote by vector \( e^I_{k(i)} \), the 2D coordinates of
the input image edge point corresponding to the \( i^{th} \) edge point of the model image,
which we denote by \( e^m_i \). Here, the index \( i \) runs over all the visible model edge points.
The aim is to minimise the distance between \( e^I_{k(i)} \) and \( e^m_i \), but first we need to find the
mapping, \( k(i) \), between the input image and the model image edge points. In other
words, given a model image edge point \( e^m_i \) we need to find which image edge point
should be put in correspondence with it. This mapping is defined by putting the model
image edge point \( e^m_i \) in correspondence with the closest input image edge point:

\[
k(i) = \arg\min_j \| e^I_j - e^m_i \| \tag{2.35}
\]

Having found the mapping, we now define the edge cost function. The aim is to max-
imise the posterior probability of the input edge map given the model and the rendering
parameters. Assuming an independent, identically distributed normal distribution for
the edge pixel errors, with zero mean and variance $\sigma_e^2$, this posterior distribution can be expressed as:

$$p(f_e|\alpha, \tau) = \frac{1}{2\pi\sigma_e^2} \prod_i \exp\left\{ -\frac{\|e'_{k(i)} - e^m_i\|^2}{2\sigma_e^2} \right\}$$ (2.36)

Note that $e^m_i$ is a function of $\alpha$ and $\tau$, but we have made this dependency implicit in the above equation for brevity. The edge cost function corresponding to the posterior distribution given in Equation 2.36 is:

$$E_e = -\ln p(f_e|\alpha, \tau) \sim \sum_i \|e'_{k(i)} - e^m_i\|^2$$ (2.37)

According to the above, fitting the edges involves, at each iteration, the following two steps: First the correspondence mapping, $k(i)$, is computed. Then, given this mapping, the model parameters are updated such that the cost function of Equation 2.37 is reduced. However, using the Chamfer Distance Transform (CDT) it is possible to unify these two steps. CDT is a mapping that associates a point of the image space, $p$, with the distance from the closest edge point:

$$CDT(p) = \min_{j=1,...,J} \|e'_j - p\|$$ (2.38)

The Chamfer Distance Transform at the image point, $p$, in essence, incorporates two pieces of information: The closest edge point to $p$, which is set in correspondence with $p$; and the distance between these two points. Hence, it replaces both the mapping $k(i)$ and the distance $\|e'_{k(i)} - e^m_i\|$. As a result, using the Chamfer Distance Transform, the edge cost function is transformed to:

$$E_e = \sum_i CDT(e^m_i(\alpha, \tau))^2$$ (2.39)

Given the input image, the CDT is calculated once at the beginning of the optimisation for all pixel locations. This defines a cost surface over the 2D coordinates of the image frame. The edge cost function is computed simply by sampling this cost surface at the positions where the model edge points are projected under the current shape and projection parameters.

The cost function of Equation 2.39 depends on the model edge points $e^m_i$, with $i = 1, ..., N_e$, which form a subset of the model vertices. Two types of model edges are considered: The textured edges result from a texture change in the inner part of the face and delimit the facial features such as the lips, eyes and eyebrows. The contour edges, on the other hand, are defined by the points that are on the border between the face area and the non-face area in the image plane.

The textured edges are formed by a constant subset of the model vertices. Variations of the shape and projection parameters do not change this subset. Therefore, it is
sufficient to annotate the set of vertices corresponding to the textured edges only once on the model, and use the same set for every fitting. The contour edges, on the other hand, depend on the shape and projection parameters. The fitting of contour edges proceeds in two steps: First, the model vertices on the contour for the current rigid parameters are selected, then these vertices are fitted to the input edges in the same way as for textured edges, i.e. using the CDT.

In order to improve the correspondence mapping between edge points, the orientation of each edge point can also be used in addition to the distance. That is, two edge points are put in correspondence only if their orientation is similar. This is done by organising the edges of the input image into four bins according to their orientation and selecting the bin with the most similar orientations for each model edge point.

Ideally, the edge detection should be such that all valid edge points are detected and no other point should be labeled as an edge. However, in practice some facial edges will be missed by any practical edge detection algorithm. Also, some non-face edges will be detected which are created by non-intrinsic factors such as illumination or noise. If an image edge point is missed, the corresponding model edge will be put in correspondence with some other image edge point which can be a valid edge from a different part of the image or a non-face edge. This can introduce a bias to the edge cost function.

In order to make the edge cost function more robust to such a scenario, an upper limit is set for the values of the edge cost surface. More specifically, the edge cost surface obtained by the Chamfer Distance Transform is thresholded such that all values higher than a given threshold are set to this threshold. The effect of thresholding the cost surface in such a way is that if a corresponding image edge point is not found within a certain distance of a given model edge, the cost for the given model edge is set to a constant value.

![Textured and Contour edges Used for fitting. Note that parts of the contour are not considered since these parts are produced by the model's boundaries and would not have equivalent edges in a realistic face image.](image)

Figure 2.6: Textured and Contour edges Used for fitting. Note that parts of the contour are not considered since these parts are produced by the model's boundaries and would not have equivalent edges in a realistic face image.
2.4. Model Fitting to a 2D Image

**Anchor Points** Similar to the SNO algorithm of [20], a set of landmarks manually annotated by the user are used. The anchor points cost function used in the MFF framework is similar to the one used in the SNO algorithm (Equation 2.28). Similar to the edge cost, the anchor point cost only involves the shape and projection parameters.

**Specular Highlights** The specular highlight feature is based on the fact that the peak of the specular lobe of the BRDF (Bidirectional Reflectance Distribution Function) has the direction of the reflection vector. That is, a facial point at the peak of the specular lobe has a normal that has the direction of the bisector of the angle formed by the light source direction and the camera direction. Hence detecting the specular highlight points yields a direct relationship between the geometry of the face surface at these points and the camera and light directions. This relationship can then be used to recover the surface normal at these points if the camera and light directions are known.

Points at the peak of the specular lobe appear very bright. If the light source intensity is strong enough, these pixels even saturate. Hence, automatically detecting them can be done with a relatively simple algorithm: If two channels of a pixel are above a threshold, say 250, then this pixel is marked as being at the peak of the specular highlight lobe. Such a point has a normal equal to:

\[
\mathbf{n}_s^*(\alpha, \tau) = \frac{\mathbf{r} + \mathbf{d}}{\|\mathbf{r} + \mathbf{d}\|} = \frac{\mathbf{v}(\alpha, \tau) + \mathbf{d}}{\|\mathbf{v}(\alpha, \tau) + \mathbf{d}\|}
\]  

(2.40)

where the vectors \(\mathbf{r}, \mathbf{d},\) and \(\mathbf{v}\) are as defined on page 21. The specular highlight cost function pushes the normal of the corresponding model points to \(\mathbf{n}_s^*\):

\[
E_s = \sum \|\mathbf{n}_{t,w}^{\xi(i)} - \mathbf{n}_s^*(\alpha, \tau)\|^2
\]  

(2.41)

where \(\xi(i)\) is the index of the triangle whose centre projects closest to the specular pixel \(i\), and \(\mathbf{n}_{t,w}^{\xi(i)}\) is the normal vector of this triangle with the superscripts \(t\) and \(w\) used to highlight that the normal vector corresponds to a triangle \((t)\) and is expressed in the world coordinates \((w)\).

The aim of the specular highlight fitting is to modify the shape parameters such that the model normal, \(\mathbf{n}_{\xi(i)}^{t,w}\), is close to the normal obtained by the specular feature, \(\mathbf{n}_s^*\). This cost function could be applied to change the vertex normal or the triangle normal. As in case of the pixel colour feature, the triangle normal is chosen due to its simpler derivative. The specular highlight feature is used at the final stage of the fitting algorithm, when the face rigid parameters and the light direction have a stable estimate. Therefore, this feature is used to update only the shape parameters, not the projection parameters or the light direction.

**Gaussian Prior** In the maximum a posteriori estimation described by Equation 2.26, \(p(\mu)\) is the prior probability distribution of the model parameters. As before, it is assumed that the shape and texture models are statistically independent. Therefore
\[ p(\mu) = p(\alpha)p(\beta) \] where the shape and texture priors are given by Equation 2.9. Hence, the corresponding cost function is:

\[ E_p = \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} \] (2.42)

**Texture Constraint** Romdhani and Vetter argued that the texture prior as given above is not restrictive enough and despite using such a prior it is possible for the texture to be over-estimated which would result in under-estimation of the light. Hence, an additional cost function was used to constrain the texture by restricting the texture values to a valid range. The cost function used for this purpose is:

\[ E_t = \sum_i (e_i)^2, \quad \text{where} \quad e_i = \begin{cases} t_i - l & \text{if } t_i < l \\ 0 & \text{if } l \leq t_i < u \\ t_i - u & \text{if } t_i > u \end{cases} \] (2.43)

In the above equation, \( t_i \) is the colour intensity of a colour channel of a model vertex used for fitting, and \( l \) and \( u \) are the lower and upper bounds of the valid intensity range, respectively.

**Light Regularisation** In addition to the above cost functions used by Romdhani and Vetter, in this work we also use an additional cost function in order to constrain the deviation of the ambient and directional light intensities from white light. The cost function used for this purpose is:

\[ E_l = \frac{1}{2} \left( 1 - \frac{L_r + L_g + L_b}{\sqrt{3} \sqrt{L_r^2 + L_g^2 + L_b^2}} \right)^2 \] (2.44)

The contribution of this cost function only vanishes when \( L_r \approx L_g \approx L_b \), where \( L_r \), \( L_g \), and \( L_b \) are the red, green, and blue channels of the ambient or the directional light respectively.

As previously mentioned, if the features are assumed to be independent, the final cost function is a sum of the individual cost functions corresponding to the respective features (Equation 2.32). Therefore, the following cost function needs to be optimised:

\[ E = \omega_c E_c + \omega_e E_e + \omega_a E_a + \omega_s E_s + \omega_p E_p + \omega_t E_t + \omega_l E_l \] (2.45)

where each \( \omega \) is a weighting constant.

Each cost function term in Equation 2.45 has different properties and is suitable for a specific purpose. Hence, Romdhani and Vetter argue that attempting to optimise this cost function in one step is not the best choice. Instead, they propose a multi-stage fitting algorithm. The algorithm is divided into different stages where each stage
fits a set of features to a set of parameters using a Levenberg-Marquardt optimisation algorithm [45]. Table 2.1 lists the five stages of the algorithm and details the features and the parameters fitted at each stage. At a given stage, a cost function is formed by the weighted addition of the cost functions of each feature fitted. For instance, the estimate yielded at stage 4 is obtained by:

$$\{\mu^*, \rho^*\} = \arg\min_{\mu, \rho} \{\omega_e E_e + \omega_c E_c + \omega_t E_t + \omega_l E_l + \omega_p E_p\}$$ (2.46)

The optimised parameters at each stage, serve as the initial parameter estimates for the subsequent stage.

**Table 2.1:** List of features and parameters fitted at each stage of Romdhani and Vetter’s MFF algorithm. In Stages 2 and 3, only the first “few” $\alpha$ and $\beta$ parameters are optimised. On the last line “segm.” stands for segmented, highlighting the fact that at this stage each segment of the model is fitted separately. (Table taken from [88])

<table>
<thead>
<tr>
<th>Stage Nb.</th>
<th>Features</th>
<th>Parameters</th>
<th>$\tau$</th>
<th>$\alpha$</th>
<th>$\gamma$</th>
<th>$\beta$</th>
<th>No. of Par.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>anchor X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>anchor X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>anchor X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>anchor X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>217</td>
</tr>
<tr>
<td>5</td>
<td>anchor X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>792</td>
</tr>
</tbody>
</table>

**Our Implementation**

The work presented in this thesis is based on the MFF framework. However, the details of our implementation differ from that of Romdhani and Vetter in the following respects:

**Light Regularisation:** The light regularisation cost presented above is not part of the original set of cost functions used by Romdhani and Vetter. We include this additional cost function in order to encourage white light.

**Colour Transformation:** For simplicity, we fix the colour transformation parameters (gains, offsets, and contrast) to default values and do not optimise them. This means that the illumination parameters to be optimised are: $\gamma = [L_r^a, L_g^a, L_b^a, L_r^d, L_g^d, L_b^d, \theta_l, \phi_l]^T$

**Parameters and Features:** Some of the parameters optimised at each stage and features used in each stage are different. the details of features and parameters fitted at each stage of our implementation are presented in Table 2.2.
Use of Anchor Points: In addition to using the anchor points feature for optimising the rendering parameters, we also use it to initialise the first three shape parameters since these parameters control the overall size of the face (see Figure 2.2). Hence, our implementation include an extra stage (Stage 2 in Table 2.2), compared to Romdhani’s implementation. This minor modification in the original MFF algorithm proved to yield more stable results.

Edge Cost: Instead of using a simple Canny edge detector we use an enhanced multi-resolution, multi-scale edge detection which is more robust. This edge detection method is detailed in Chapter 4.

Translation in Z: Since the camera focal length ($f$) and the translation along the Z axis ($t_{w,z}$) have similar effects, we only optimise the focal length and fix $t_{w,z}$. Therefore we only have 6 projection parameters: $\tau = [\theta_x, \theta_y, \theta_z, t_{w,x}, t_{w,y}, f]^T$.

Illumination Model: The Phong reflectance model discussed in Section 2.3.3 is computationally expensive since one needs to continually recalculate the dot product, $\langle r_i, v_i \rangle$, between the vector of viewing direction, $v_i$, and the vector of light reflection direction, $r_i$.

Instead, we use a modification of this model known as the Blinn-Phong reflectance model [21]. To this end, a halfway vector, $h$, is calculated between the viewing direction, $v$, and the light direction vector, $d$:

$$h = \frac{d + v}{\|d + v\|} \quad (2.47)$$

The dot product $\langle r, v \rangle$ in Equation 2.13 can then be replaced by $\langle n, h \rangle$, where $n$ is the normalised normal vector at the given point.

Figure 2.8 shows two sample fitting results using our implementation of the MFF algorithm on inputs from the XM2VTS dataset [74].

Table 2.2: List of features and parameters fitted at each stage of our implementation of the MFF algorithm. In stage 2, only the first 3 shape parameters are optimised using the anchor points. In Stages 3 and 4, the first 20% of the $\alpha$ and $\beta$ parameters are optimised. Considering that we have 55 shape and 123 texture parameters, this means that in Stage 3, only the first 11 shape ($\alpha$) parameters are optimised and in Stage 4, only the first 25 texture ($\beta$) parameters. On the last line “segm.” stands for segmented, highlighting the fact that at this stage each segment of the model is fitted separately.

<table>
<thead>
<tr>
<th>Stage Nb.</th>
<th>anchor</th>
<th>textured edges</th>
<th>contour edges</th>
<th>pixel colour</th>
<th>texture constr.</th>
<th>specular highl.</th>
<th>light</th>
<th>prior</th>
<th>$\tau$</th>
<th>$\alpha$</th>
<th>$\gamma$</th>
<th>$\beta$</th>
<th>No. of Par.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>all</td>
<td>all</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>segm</td>
<td>712</td>
</tr>
</tbody>
</table>
2.5 Texture Extraction

After the fitting process has been completed, the fitted model together with the rendering parameters can be used to extract facial texture from the input image in order to have the option of using original texture, where available, instead of the texture reconstructed by the model. Tena et al. [86] proposed a texture mapping algorithm to serve this purpose using the CVSSP-3DMM. The texture mapping algorithm extracts facial texture from the input image and stores it in a 2D texture map.

The first step is to create a texture map where the texture values corresponding to each point of the model are stored. A 2D map on which all triangles of the model are displayed with their proportions preserved is required so that the colour value for every point in each triangle is stored in the corresponding location. The problem is to find a 2D representation of the generic face model that preserves the geodesic\(^4\) distance between its vertices. Such a representation is found using the isomap algorithm proposed by Tenebaum et al. in [100]. The algorithm takes as input the distances \(d_T(i,j)\) between all pairs \(i, j\) from \(N\) data points in the high dimensional space \(\Upsilon\). The output is a set of vectors \(\omega_i\) in a lower dimensional Euclidean space, \(\Omega\), that best represents the intrinsic geometry of the input data. The result of applying the isomap algorithm to the generic model is the flattened mesh shown in Figure 2.7.

![Generic Model](image1.png) ![Iso-generic Model](image2.png)

Figure 2.7: The generic model and the flattened model using isomap dimensionality reduction (iso-generic model). Image courtesy of [86].

The flattened generic model, referred to as iso-generic model, can now be overlaid with a rectangular grid of pixels. This grid defines an image that can be parametrised by a 2D coordinate frame to create a texture map for the model. Each triangle of the model is represented on this new image. After the fitting process is completed, if a triangle is visible in the input image, the RGB values of the pixels of the triangle are taken from the input image and copied to the location that corresponds to that same triangle in the newly created texture map. The pixels in the new texture map that correspond

\(^4\)The shortest path connecting two points on a surface.
to triangles that are not visible in the image are filled using the RGB values of the corresponding triangles of the optimised morphable model, i.e. reconstructed texture. Figure 2.8 shows two exemplar fitting results together with the texture extracted from the inputs in each case. The texture maps on the bottom row illustrate how the texture extraction algorithm creates a texture map combining original image information with that reconstructed by the morphable model. Note that for the input on the right, the subject is wearing glasses which are not modelled by the 3DMM; however, the model still manages to fit the face using information from other areas of the face. The glasses are present in the texture map (bottom row) since this is the original texture extracted from the input image.

Figure 2.8: Sample fitting results on the XM2VTS dataset images. Top row: input image, middle row: fitted model super-imposed on the original image, bottom row: texture map extracted from the input.
2.6 Face Recognition Using a 3D Morphable Face Model

As mentioned earlier, a 3D morphable face model can be used in many different applications such as face rendering, facial expression transfer etc. However, in this thesis we are mainly concerned with the application of 3DMM in face recognition. There are two main scenarios in which a 3DMM can be used for the purpose of face recognition.

**Image-Based Approach:** In this scenario, which we also refer to as “3D-assisted face recognition”, the 3DMM is used as a means to normalise the input images. In other words, it is used to normalise the pose and illumination to a default setting (e.g. frontal face with uniform frontal illumination). Given an input image with arbitrary pose and illumination, a 3DMM is fitted to this image to extract the model shape and texture parameters as well as the rendering parameters that correspond to the given input. One can also use the fitted model to extract the facial texture, where available, from the input image, as described in Section 2.5. Using the information obtained from the input image by the model, a new normalised face of the subject is then rendered using the default rendering setting which corresponds to the normalised pose and illumination. This image is then used for recognition using a conventional 2D face recognition engine which is trained and optimised for the normalised pose and illumination.

**Model-Based Approach:** In this scenario, the shape and texture parameter vectors extracted from the input image are used to build a feature vector describing the input face. The feature vectors of the gallery images are stored as the gallery set. For each probe comparison, the model is fitted to the given probe image and the feature vector corresponding to the probe image is compared to the gallery feature vectors for recognition. A similarity (or dis-similarity) measure is calculated between the probe feature vector and each gallery feature vector in order to identify the best match among the gallery set (identification) or verify whether it passes a certain acceptance threshold (verification).

Figure 2.9 illustrates diagrams for these two scenarios. In the experiments of Chapter 5, the model-based approach is used while in Chapter 6 the image-based approach is used for face recognition.

2.7 Discussion and Conclusions

This chapter covered part of the necessary background for the work presented in this thesis. We reviewed the 3D Morphable Face Model as a tool which can be used for pose- and illumination-independent facial analysis. We described how such a model can be built from a set of exemplar 3D facial scans and we reviewed the main algorithms which can be used for fitting such a model on a 2D image; namely, the SNO and the MFF algorithms. Both of these algorithms perform the fitting using a MAP framework and aim to find the maximum posterior probability of the model parameters together.
with a set of rendering parameters, given the input 2D image (as well as some features extracted from this image in the case of MFF).

Finally, we described how a 3DMM can be used for face recognition. We discussed two approaches for this task which have both been used previously in the literature and which we will also use in the succeeding chapters of this thesis.

As discussed in Chapter 1, the challenge of a face recognition system is to separate the effects of the intrinsic and extrinsic factors which affect a facial image in order to yield a robust recognition. In fact, this is exactly what the 3DMM aims to do: modelling the shape and texture of a face (intrinsic factors) independently of extrinsic factors such as pose and illumination. The use of 3DMM in modelling facial expressions has also been investigated in the literature (e.g. in [4]). Thus, the 3DMM provides a powerful tool for unconstrained face recognition under varying poses, illuminations, and expressions. However, to the best of our knowledge, the 3DMM has not been successfully applied to face recognition in low-resolution images. As we will see in Chapter 4, the fitting criteria commonly used for model fitting become sub-optimal in low-resolution images. Therefore, attempting to fit a 3D model to LR face images using the conventional approaches described in this chapter is subject to large errors.

Chapter 3 of this thesis will focus on LR face recognition and present a review of the relevant literature. We will return to the 3DMM fitting in Chapter 4 and show why the conventional fitting algorithms are not suitable for modelling low-resolution faces. We will then present a resolution-aware model fitting algorithm by using some of the ideas reviewed in Chapter 3, and present a comparison of our model fitting approach with the conventional MFF approach. Our resolution-aware model fitting algorithm paves the way towards extending the use of 3D morphable models to an application in which they had not been successfully used previously: low-resolution face recognition.
2.7. Discussion and Conclusions

(a) Image-Based Approach

(b) Model-Based Approach

Figure 2.9: Two scenarios for face recognition using a 3D Morphable Face Model.
Chapter 3

A Review of Low-Resolution Face Recognition Techniques

3.1 Introduction

We mentioned in Chapter 1 that one of the extrinsic sources of variation in a facial images is the image resolution, and that the performance of a facial recognition system would degrade due to low-resolution input images. It can usually be assumed that high quality images taken under controlled conditions are available at enrollment in order to build a suitable gallery set. Thus, the problem of low-resolution (LR) face recognition normally refers to the case where the resolution of the probe image is low, whereas the gallery set is high-resolution (HR). For instance, in images captured by CCTV cameras faces are often captured with low resolution due to distance from the camera and/or large camera focal length causing blurring. Another example is in automatically organizing family photographs where faces might be small in some pictures.

It has been shown in various studies (eg. [106]) that the performance of a face recognition system can considerably degrade in such a situation. This is due to the fact that a low-resolution image lacks a substantial amount of descriptive information which is necessary for distinguishing faces. Furthermore, the features used for describing the face would differ if they were extracted from an LR or an HR image. Therefore, if the resolutions of the gallery and probe differ considerably, the set of features used for describing them would also be considerably different which results in a degraded performance of the recognition system.

Many researchers have attempted to address the problems posed by low resolution over the past years. Most of the approaches used in the literature for LR face recognition can be categorised into one of the following four categories, based on the strategy they chose to address the problem:

**Training in LR:** One strategy is to reduce the resolution of the gallery set to that of the probe image and train a new system in LR. This way the problem of mismatched feature extraction due to mismatched resolution can be addressed.
However, many descriptive details about the difference between one person and another can only be captured in images of sufficient resolution and an important part of the performance degradation is due to the lack of such detailed information in LR images. This problem still remains if the gallery set is reduced to LR. Furthermore, this is not a practical approach since for each given probe image, a new gallery set must be generated, and feature extraction and classifier training must be repeated which makes this a daunting and impractical approach. However, there have been examples of rather successful uses of such strategies. For instance, in a recent work, Shekhar et al. [96] utilised such an approach where they use a training set and extend it by generating new images with different illuminations. The extended training set is then down-sampled to the probe resolution and used to learn a resolution-specific dictionary which is then used for recognition of the LR probe image. They report relatively good results using this approach. However, the problem of having to generate a training set based on the resolution of the input still exists.

**HR reconstruction:** Another strategy is to reconstruct a high-resolution representation of the probe image using one or multiple LR probe images; a task known as super-resolution (SR). Many different approaches to super-resolution have been proposed in the literature ranging from general image super-resolution to methods developed specifically for face super-resolution. The aim here is to reconstruct or estimate a high-resolution image of the face captured by the probe image(s) such that it contains a similar amount of descriptive information as the HR gallery set. Thus, super-resolution is used as a pre-processing step before recognition.

**Direct HR-LR matching:** While the previously mentioned strategies aimed at matching the resolutions of the gallery and probe images prior to recognition, a third category of approaches has recently emerged where the aim is to avoid such pre-processing steps and attempt to match the HR gallery and LR probe images directly. This is done in a number of fairly recent works by either combining the SR and recognition tasks into a simultaneous framework ([47]), or by mapping both the HR and LR images into a common space where the discriminatory information used for face matching is less affected by the difference in the resolutions of the original images ([17], [113]).

**Model-based approaches:** The 3D Morphable Model described in Chapter 2 is only one example among various models used for face recognition in the literature. Other examples of face models include Active Shape Model (ASM) [27] and Active Appearance Model (AAM) [30], with the later drawing relatively more attention. In general, the model-based approach to face recognition usually includes a model fitting process to describe the facial images in terms of a number of model parameters which can then be passed to a recognition engine. Following the successful application of face models to the problem of face recognition under varying poses, illuminations, and expressions, attempts have been made to extend their application to the case where the image resolution is low.

In the remaining parts of this section, we will elaborate on the latter three categories of approaches. We review a number of super-resolution approaches in the literature and
discuss their advantages and disadvantages in the context of unconstrained face recognition in Section 3.2, while Section 3.3 reviews some approaches to LR face recognition which use the direct HR-LR matching approach. Section 3.4 will discuss model-based approaches to the problem of LR face recognition, and Section 3.5 will conclude this chapter with a discussion about the shortcomings of the reviewed methods. Through the review presented in this chapter and the discussion of the presented methods, we will elaborate on the motivation for our work which will be presented in the following chapters.

![Figure 3.1: Conceptual illustration of the main categories of approaches used in the literature for LR face recognition. The red arrows show the path for the “Training in LR” strategy, the HR gallery is subsampled to create an LR gallery. Matching is then performed between LR images. The black arrows show the path for the “HR Reconstruction” strategy where an HR probe image is reconstructed and compared to the HR gallery. The green arrow shows the “Direct HR-LR matching” path and the blue arrows show the model-based strategy in which a model is used to describe both the HR gallery and the LR probe image through the fitting process.](image)

### 3.2 Super-Resolution

As a solution to the problems posed by low-resolution inputs, super-resolution techniques are used to increase the resolution of the probe face prior to recognition. In general, super-resolution is the process of recovering a high-resolution representation of a scene given one or more images of the same scene. Many super-resolution methods exist in the literature which can be largely categorised into reconstruction-based and example-based methods. The reconstruction-based methods aim to fuse information
from multiple different observations (low-resolution images) to recover the missing information and build an image with a higher resolution than any of the inputs. On the other hand, example-based methods utilise a set of exemplar images to model or learn the relationship between low- and high-resolution images and use this as an additional source of information. Unlike reconstruction-based approaches, the example-based methods are often applicable to cases where only one input observation is available. The example-based approaches can further be categorised into generic approaches and class-specific approaches which focus on super-resolving images that belong to a specific class of images (e.g. text or human faces).

In the following, representative methods of super-resolution will be reviewed. We start our discussion with a model for the image formation process which is commonly used by most SR methods in order to describe an LR image in terms of the underlying scene or the sought HR image. We then continue by reviewing some reconstruction-based and example-based methods. Finally, a number of methods for super-resolution of a specific object, namely the human face, are presented.

### 3.2.1 Observation Model

The resolution of an image depends on the physical characteristics of the camera: the characteristics of the optical system as well as the density and spatial response of the CCD elements. Naturally, the first step to super-resolution would be to understand and model the processes in which an image is formed and affected by these characteristics. This section formulates the image formation process in a generic observation model relating an “original” high-resolution image to the observed low-resolution images. The model derived in this section is commonly used in the literature with minor modifications or simplifying assumptions (e.g. [54, 7, 8, 10, 56]).

Let \( h \) denote the sought high-resolution image, represented as a vector by stacking its columns together, and \( l_k \) be the \( k^{th} \) low-resolution image observed \( (k = 1, 2, ..., K) \); also represented as a vector. It is common practice in the super-resolution literature to assume that all low-resolution images are observations of the same underlying high-resolution image. In other words, \( h \) remains constant during the acquisition of all low-resolution images. A low-resolution image is then considered as the result of warping, blurring, and down-sampling of the unknown high-resolution image. Different low-resolution images differ from one another in geometric transformations, blurring, image quantisation, and noise. Representing the images as vectors, the observation model can be represented in matrix form as:

\[
l_k = D_k B_k W_k h + n_k \quad \text{for} \quad 1 \leq k \leq K
\]

where \( D_k, B_k, \) and \( W_k \) are matrices representing the down-sampling, blurring, and warping associated with the \( k^{th} \) image, respectively. Also, \( K \) is the number of low-resolution observations, and \( n_k \) is an additive noise term.

Any geometric registration relating the \( k^{th} \) low-resolution image, \( l_k \), to the unknown high-resolution image, \( h \), is represented by the warping matrix \( W_k \). This information
is generally unknown. Thus, in general, the registration information is estimated with respect to one of the low-resolution frames, and the coordinate frame of the HR image is assumed to be the coordinate frame of this reference LR image, enlarged by the appropriate enlargement factor. Two general trends exist in the literature. The first is to estimate registration prior to super-resolution so that in the super-resolution stage, the registration parameters are assumed to be known (e.g. in [8]). The second approach is to estimate both the registration parameters and the high-resolution image simultaneously (simultaneous registration and super-resolution; e.g. in [102]).

Blurring can be due to external causes such as motion blur, out-of-focus blur etc. or causes relating to the imaging device, i.e. the point spread function (PSF) of the camera. The PSF-related blurring can further be decomposed into two parts representing the blurring caused by the optics, and the blurring caused by spatial integration performed by the CCD sensor. The blur can be modelled by convolution with a low-pass kernel. In most practical cases the blurring operator cannot be obtained. Thus, if explicit representation of the blurring kernel is required, it is usually estimated.

Finally the down-sampling operator, $D_k$, generates a low-resolution image from the warped and blurred image. The noise term in equation 3.1 accounts for any deviation of the observed pixel value from its expected value. These deviations can be caused by many sources such as measurement error (e.g. quantisation error) or model error (e.g. error in registration or blurring kernel estimation). For practical reasons, this noise is usually assumed to be normally distributed with zero mean. This choice seems plausible since having many possible sources of noise which are independant, the central limit theorem would suggest that the additive noise in equation 3.1 can be expected to have a “close-to-normal” distribution.

Although the observation model described here can be formulated in a more specific manner (e.g. by considering different terms for optical and sensor-related blur), in the super-resolution literature it is common to use an observation model similar to equation 3.1, since it is sufficient for the needs. In fact, it is common practice (e.g. see [13][25][55]) to concatenate the three matrices $D_k$, $B_k$, and $W_k$ into a single matrix $M_k$ which includes warping, blurring, and down-sampling operations, and represent the observation model as:

$$l_k = M_k h + n_k$$ \quad \text{for} \quad 1 \leq k \leq K \quad (3.2)$$

In the above equation, each element of the vector $l_k$ represents a pixel of the $k^{th}$ LR image. The value of this pixel is given as a weighted sum of the HR pixels (elements of the vector $h$) plus an additive noise (the corresponding element of the vector $n$). The weights of the weighted sum are elements of the corresponding row in the matrix $M_k$. The weights determine the contribution of the HR pixels to the LR pixel at hand. Naturally, only the HR pixels which have some overlap with the given LR pixel after appropriate warping would contribute to it. The rest of the HR pixels which do not overlap with the LR pixel at hand would not contribute to it; hence have zero weight.
Accordingly, an equivalent representation for the imaging model of Equation 3.2, is:

\[
L_k(m,n) = \sum_{(p,q) \in \text{bin}(m,n)} w(p,q)H(p,q) + N_k(m,n) \quad \text{for} \quad 1 \leq k \leq K
\] (3.3)

In this Equation, \((m,n)\) indexes the pixels in an LR image and \((p,q)\) indexes the pixels of the HR image. Thus, \(L_k(m,n)\) is the \((m,n)^{th}\) pixel of the \(k^{th}\) input LR image, \(H(p,q)\) is the \((p,q)^{th}\) pixel of the HR image, and \(w(p,q)\) is the weight associated with it. Also, \(N_k(m,n)\) is the additive noise corresponding to the pixel \(L_k(m,n)\), and the sum is taken over \(\text{bin}(m,n)\) which represents all HR pixels which have some overlap with the given LR pixel after appropriately warping the HR image.

The weights of this weighted sum depend on the blurring kernel, as well as the amount of overlap between the HR and LR pixels, which in turn depends on the warping.

### 3.2.2 Reconstruction-Based Methods

Reconstruction-based methods typically make use of the available information from multiple low-resolution observations to reconstruct a high-resolution image by trying to invert the process of image formation discussed in the previous section. This typically includes registration to invert the warping, interpolation to represent the image information on a uniform high-resolution grid, and restoration to remove noise and blurring. A simple approach would be to densely register the low-resolution images to a single high-resolution grid and merge the information from different images to reconstruct a higher resolution image. Such a scenario was used in the early work of Ur and Gros [104] where the input images are assumed to undergo relative shifts which are known exactly \textit{a priori}. After interpolating and merging the input images onto a finer grid, a blurred image is obtained which contains the information from multiple observations. This image is then deblurred using a kernel derived from the inverse of the blur operator.

More sophisticated methods mostly utilise the super-resolution reconstruction \textit{constraint}: the super-resolution image, when appropriately warped and down-sampled to mimic the image formation process, should yield the low-resolution input images. Assuming a single LR input, satisfying this constraint can be thought of as finding the minimiser of \(\|Mh - l\|^2\), where \(M\) is an observation model similar to the one described by Equation 3.2. In many cases, this does not have a unique solution. Hence, constraints such as smoothness of the final result are often introduced on the solution. A simple way to introduce such constraints is by using Tikhonov Regularisation (see [114], for example).

One of the first methods to use the super-resolution constraint was proposed by Peleg \textit{et al.} [81]. They assumed that the imaging process is known and that the low-resolution images have been registered with sub-pixel accuracy. Starting from an initial guess for the unknown high-resolution image, the imaging process is simulated to generate a set of simulated low-resolution images resembling the low-resolution inputs. An error is then defined between the original and simulated low-resolution images. By minimising this
error with respect to high-resolution pixels iteratively, the final high-resolution image is obtained. The error between the observed and simulated low-resolution images is defined as:

$$ E^j = \sum_k \sum_{(m,n)} |\hat{L}^j_k(m,n) - L_k(m,n)| $$

where $E^j$ is the error of the $j^{th}$ iteration, $L_k$ is the $k^{th}$ observed low-resolution image, $\hat{L}^j_k$ is the estimate for the $k^{th}$ low-resolution image simulated at the $j^{th}$ iteration.

Irani and Peleg [54] presented an iterative back-projection (IBP) super-resolution approach in which the error between the observed low-resolution images and the low-resolution images estimated by the observation model is back-projected to update the estimate for the high-resolution image in each iteration. By iteratively updating the high-resolution image, the energy of the error is minimised. The iterative update scheme to estimate the high-resolution image is expressed as:

$$ H^{j+1}(p,q) = H^j(p,q) + \sum_{m,n \in \bigcup_k Y^{p,q}_k} [L_k(m,n) - \hat{L}^j_k(m,n)] \cdot W(m,n;p,q) $$

where $H^j$ is the high-resolution image estimate at the $j^{th}$ iteration, $L_k$ is the $k^{th}$ low-resolution observation, and $\hat{L}^j_k$ is an estimate of the $k^{th}$ low-resolution image obtained by applying the appropriate observation model to $H^j$. Furthermore, $Y^{p,q}_k$ is the set of LR pixels that are influenced by the high resolution pixel $(p,q)$.

In Equation 3.5, $W(m,n;p,q)$ is a weighting term, and is given by:

$$ W(m,n;p,q) = \frac{\psi^{BP}(m,n;p,q)^2}{\sum_{\hat{m},\hat{n} \in \bigcup_k Y^{p,q}_k} \psi^{BP}((\hat{m},\hat{n});p,q)} $$

where $\psi^{BP}$ is a back-projection kernel, and $c$ is a normalising constant.

There can be more than one $\psi^{BP}$ leading to convergence. However, the choice of $\psi^{BP}$ can influence the characteristics of the solution when there is more than one possible solution. Hence, $\psi^{BP}$ can be utilised as a constraint so that the final solution has some desired property, e.g. smoothness.

The work of Irani and Peleg [54] assumes Euclidean transformation (translation and rotation) for registration. Their approach was extended by Mann and Piccard [70, 71] to a full projection transform. Although the IBP method provides a straightforward and intuitive approach to super-resolution, it does not allow for incorporating prior knowledge into the framework and the solution may not be unique due to the ill-posed nature of the problem.

An alternative approach proposed by Stark and Oskoui [97] and later developed further by others ([98, 80, 38]), is based on projection onto convex sets (POCS). This approach is an iterative method that allows for incorporation of prior knowledge into
Chapter 3. A Review of Low-Resolution Face Recognition Techniques

the super-resolution framework. Assume that the solution is required to have a number of properties based on the prior knowledge. A closed convex set, $C_i$, is associated with each of the desired properties where each $C_i$ is the set of vectors that satisfy the $i^{th}$ property. Assuming that the sets $C_i$ have a non-empty intersection, the final solution lies in the intersection of all sets. The central theorem of the method is as follows: denote the vectorised high-resolution image at the $j^{th}$ iteration by $h_j$ and assume there exists a projection $P_i$ that projects an arbitrary vector (image) to the set $C_i$. Then the recursion:

$$h_{j+1} = P_m P_{m-1} \ldots P_2 P_1 h_j \quad (3.7)$$

where $h_0$ is an arbitrary starting point, converges weakly to a feasible solution that lies in the solution set $C_s = \cap_{i=1}^m C_i$ which is also a closed convex set.

Although this may not be a trivial task, it is in general easier than finding a projection that projects $h_0$ to $C_s$ in one step [97]. Based on the observation model, a data consistency constraint is defined as a set of images which satisfy the super-resolution reconstruction constraint. Furthermore, additional constraints that are imposed by the prior knowledge can be added. For instance, Stark and Oskoui [97] used an amplitude constraint, an energy constraint, a constraint on the permitted rms deviation from some known reference image, and a bounded-support constraint.

Other extensions to the POCS approach have been proposed in the literature [98, 80, 38] which include frameworks to consider many generalised conditions such as space-varying blur, nonzero physical dimension of CCD elements, different types of noise, arbitrary sampling latices, and multiple moving objects in the scene. Although the POCS approaches have the advantage of incorporating the a priori information, they have the disadvantages that their solution is not unique, convergence is slow, and computational cost is high.

Another framework for incorporating a priori knowledge in the super-resolution solution are the stochastic approaches known as Bayesian approaches. In Bayesian approaches, the posterior probability distribution function (PDF) of the sought high-resolution image given the low-resolution observations is maximised with respect to the high-resolution image pixel values. These approaches are known as maximum a posteriori (MAP) estimation:

$$\hat{H} = \arg\max_H p(H \mid L_1, L_2, \ldots, L_K) \quad (3.8)$$

where $\hat{H}$ is the sought high-resolution image and $L_k$ is the $k^{th}$ low-resolution observation. According to the Bayes rule, equation 3.8 can be expressed as:

$$\hat{H} = \arg\max_H \{p(L_1, L_2, \ldots, L_K \mid H)p(H)\} \quad (3.9)$$
or equivalently:

$$
\hat{H} = \arg\min_H \{-\ln(p(L_1, L_2, \ldots, L_K \mid H)) - \ln(p(H))\}
$$

(3.10)

In the above equations $p(L_1, L_2, \ldots, L_K \mid H)$ is the likelihood of the LR observations, given the sought HR image, while $p(H)$ is the prior probability distribution of the high-resolution image before observing the low-resolution images. The likelihood term is defined using the low-resolution observations and the super-resolution constraint. The prior term can be used to constrain the solution according to prior knowledge about the high-resolution image in order to obtain a regularised solution.

As a special case of MAP estimation, maximum likelihood (ML) estimation has also been applied to the image super-resolution problem. ML estimation can be considered as a special case of MAP where the prior term, $p(H)$, is effectively ignored.

Tom and Katsaggelos [102] used an ML approach for simultaneous registration and super-resolution. Although the ML method improves the input images, for large magnification factors the estimation becomes highly ill-conditioned and the solution is highly sensitive to noise in the input observations, the parameters of the generative model, and the registration transformation. Attempts to overcome this problem include placing hard constraints on the individual pixel intensities or use of a Bayesian prior model of the super-resolved image by modelling the image as a first-order stationary Markov Random Field (MRF) and including a spatial prior to model the spatial dependencies of neighbouring pixels. This would result in a maximum a posteriori (MAP) estimation.

Such priors are typically either chosen to impose some kind of smoothness on the image (e.g. Gaussian MRF) or have some edge preserving characteristics (e.g. Huber-MRF). Schulz and Stevenson [92] proposed a discontinuity preserving MAP reconstruction method using the Huber-Markov Gibbs prior for super-resolution of LR video sequences. A joint MAP estimation for simultaneous registration and super-resolution was proposed by Hardie et al. [46] which uses a Gaussian prior.

Humbolt and Djafari [52] proposed a method in which the image is assumed to be composed of a number of homogeneous regions and the pixels inside each region are assumed to follow a given probability law. This method performs super-resolution by hypothesising the properties of the pixel distributions in each region (prior probabilities).

Although some reconstruction-based approaches can also be used for de-blurring a single image (e.g. [54]), they do not add any additional information to the image through de-blurring since they mainly rely on only one source of information: the super-resolution reconstruction constraint. Hence, the reconstruction-based super-resolution methods are in general incapable of performing single-frame super-resolution. Another disadvantage of reconstruction-based methods, as shown by Baker and Kanade [9] is that even in the case of multiple low-resolution observations, the reconstruction constraints provide far less useful information as the magnification factor increases. They showed that for large enough magnification factors, any smoothness prior leads to overly smooth results with very little amount of high-frequency information regardless of the number
Chapter 3. A Review of Low-Resolution Face Recognition Techniques

of low-resolution observations used. This means that there is effectively a theoretical limit to how well reconstruction-based methods can perform. Lin and Shum [66] further investigated this problem and derived explicit limits for the magnification ratio of reconstruction-based super-resolution methods under the assumption that the registration of LR images is locally translational.

A remedy to the above-mentioned limitations of reconstruction-based super-resolution methods is to use additional sources of information as well as the low-resolution observations and the reconstruction constraint. This leads to another generation of super-resolution methods, namely example-based methods, which will be discussed in the next section.

3.2.3 Example-Based Methods

A rather more recent approach to image super-resolution is the example-based approach in which an additional source of information is used to extend the limits of reconstruction-based super-resolution.

Collections of image pixels are special signals that have much less variability than a set of completely random signals. A number of methods have tried to exploit these regularities to infer plausible image information. These methods generally use a number of exemplar images as training images to learn or model the relationship between the low-resolution and high-resolution image pairs or to learn more appropriate prior terms. Freeman et al. [43, 44] proposed a patch-based approach that used a set of training images to learn the relationship between a sharp image and its low-resolution counterpart. They used a Markov network to probabilistically model the relationship between the HR and LR patches. Figure 3.2 shows the Markov network in which $H_i$ and $L_i$ represent the $i^{th}$ patch of the HR and LR images, respectively. By blurring and downsampling a set of training HR images, the authors generated a training set of sharp and blurred image pairs which is used to learn the compatibility functions of this network ($\phi$ and $\psi$). This approach (named VISTA - Vision by Image/Sequence TrAining- by the authors) learns the prior from a set of examples instead of hypothesising it (as in the reconstruction-based method of [52], for example).

In [84], Pickup et al. presented an example-based approach that uses a MAP framework. Assuming an independent, identically distributed (i.i.d) Gaussian distribution for the pixel noise, the likelihood term is defined using an observation model similar to the one discussed earlier in this section. The prior term is determined using a set of exemplar images. For each pixel of the sought HR image, the prior is defined by selecting a pixel from the high-resolution database that is most similar to the pixel at hand. The prior is then defined as a Gaussian distribution centred at the grey value of this winner pixel.

Although none of the above approaches assume that the training and the input images belong to the same class of images, they can generally produce visually pleasing results. However, [43, 84] have included examples of cases where the input image was super-resolved using a training set that included completely irrelevant images (e.g. super-resolution of an image of leaves using a sample set consisting of text images; see Figure 3.3) and have shown that the results are not acceptable. This shows the importance of
3.2. Super-Resolution

Figure 3.2: The Markov model used by Freeman et al. (Figure courtesy of [43, 44])

the sample set in the example-based approaches and confirms the fact that the sample set must represent the visual content of the set of input images. As Freeman et al. stated in [43], “without restriction to a particular class of images, it is unreasonable to expect to generate the correct high-resolution information.” However, if the application allows us to assume that the images belong to a specific domain or object (such as text, or faces) example-based learning can considerably improve the results since the visual content of the sample set can be selected such that it represents the object of interest. An example of such a case is face super-resolution where the object of interest is a human face and the methods and sample images are tailored for application to images of a human face.

3.2.4 Face Super-Resolution

Many different approaches have been proposed for super-resolution of specific objects of interest that belong to a particular domain of images (e.g. text or human face). In these methods, the training and input images belong to the same domain of images and contain the same (or similar) objects. Using exemplar images of the same domain as the inputs can provide information of finer detail and pixel distribution properties for super-resolution learning and inference. In this section we will review some methods dedicated to facial image super-resolution. Since the problem is now restricted to facial images, the training samples will also be facial images.

Wang and Tang [107] proposed an eigentransformation approach in which the input image is represented as a linear combination of the training LR images using Principal Components Analysis (PCA). The HR image is generated by replacing the LR eigenimages with their HR counterparts while keeping the mixture coefficients. Since the coefficients are not computed from the HR data, some non-face-like distortions appear which are then reduced by projecting this HR image to an HR eigenface space (calculated from the training HR images) and putting some constraints on the principal components.

Liu et al. [67] proposed a two-step method for face super-resolution. This method
is based on the fact that the HR image is a combination of common (global) face properties and individual (local) characteristics. The global and individual properties are captured by a global parametric model ($H^g$) and a local non-parametric model ($H^l$) respectively. The final HR image ($H$) is assumed to be a combination of these two models: $H = H^g + H^l$. In this framework, $H^g$ is obtained by modelling a set of HR training images using PCA, and defining $X^*$ as the set of PCA coefficients that maximises $p(L|X)p(X)$. The solution for $X^*$ can be given analytically which makes this step very fast. In the second step, $H^l$ is found using an MRF model. Each patch in $H^l$ is defined in the MRF network by its neighbouring patches in $H^l$ and the corresponding patch in $H^g$. An energy function is defined for the network as the sum of two external and internal terms. The external term represents the connectivity statistics between corresponding patches in $H^g$ and $H^l$ while the internal term describes neighbouring statistics between patches inside $H^l$. The optimal patch is found as the patch that minimises this energy. Li and Lin [65] have further improved this approach by applying PCA to both LR and HR images in the first step in order to incorporate the estimation of noise model and also using a MAP framework instead of the MRF model for finding the optimal local face.

In [25], Capel and Zisserman presented another example-based, domain-specific method for faces. Each face image is broken down into 6 regions and each region is represented with PCA. The PCA representation is then used in 3 different ways for super-resolution. The first method is a maximum likelihood (ML) estimation of the super-resolved image in which the PCA is used to constrain the solution to lie in the face sub-space spanned by the PCA components. The second and third methods are both MAP estimations. In the second approach, again the solution is constrained to lie in the face sub-space and
the prior term is defined over the face sub-space imposing a penalty on the Mahalanobis
distance of the solution from the mean. However, the third method does not constrain
the solution to the face sub-space. Instead, it encourages the solution to lie near the
PCA sub-space and the prior is defined over the whole image space.

A very well-known approach for object-specific super-resolution is Face Hallucination\(^1\)
(also named as the reconstruction method) proposed by Baker and Kanade [7, 8, 10].
This approach is applicable to one or more input images. In the case of multiple input
images, the geometric registration transformation is assumed to be only translational
and the exact translations are assumed to be known a priori. This approach uses a
MAP framework. The likelihood term is defined using an observation model similar to
equation 3.2. The prior term is a recognition-based prior which is defined on the
gradients of the HR image using “recognition” results of some generic local “features”.
Recognition - in this context - means finding the most “similar” pixel from the training
set for each pixel of the LR input image(s). The set of all possible values for the
input LR images are assumed to be partitioned by the recognition results into a set of
sub-classes. The prior term is then given as: \(p(H) = \sum_j p(H | L_k \in C_j) \cdot p(L_k \in C_j)\)
where \(L_k\) is the \(k\)th input LR image and \(C_j\) is the \(j\)th sub-class. Once the subclass is
determined by the recognition results, the prior effectively simplifies to \(p(H | L_k \in C_j)\).
This class-specific prior is then defined such that it encourages the gradient of the
super-resolution image to be close to that of the closest matching training samples.

The Name reconstruction for this method is a combination of the terms recognition and
reconstruction which suggest the two main sources of information for the method: the
reconstruction constraints and the recognition-based prior term.

Recently, Bilgazyev et al. [13] proposed an approach for face super-resolution in which
high-frequency components of facial images are learnt and applied to the LR input
in order to create the SR image. Their approach consists of two stages. In the first
stage, a Dual-Tree Complex Wavelet Transform is used to extract the high-frequency
components from a database of facial images. This information is then used in the
second stage to create an SR image for a given LR input. Selection of the high-
frequency components is based on similarity between the input image and images from
the training set.

The above face super-resolution methods generate an HR image of a single facial modality
(i.e. at a fixed expression, pose, and illumination). In [56] Jia and Gong have
presented a generalised face super-resolution method which aims at multimodal super-
resolution of faces. In other words, given an LR image at a specific modality, it aims
at generating HR images of the same, as well as other different modalities. A
tensor structure is used to incorporate information and interactions of (training) images
of multiple modalities at different resolutions. Considering only two modalities as an
example, the HR images \(H_1\) and \(H_2\) can be generated by maximising:

\[
p(L_1 | \hat{L}_1)p(\hat{L}_1)p(\hat{L}_2)p(\hat{L}_2)p(H_1^{lm}|\hat{L}_1)p(H_1^{lm}|\hat{L}_2)p(H_2^{lm}|\hat{L}_1)p(H_2^{lm}|\hat{L}_2)
\]

\(1\)Although the term Face Hallucination was first coined for this specific method, it is occasionally
also used in a more general sense in the literature to refer to (example-based) face super-resolution
In equation 3.11, $L_1$ is the input LR image at modality 1, $H_1$ and $H_2$ are the sought HR images at the input and one other modality, respectively, while $\hat{L}_1$ and $\hat{L}_2$ are their corresponding LR images to be synthesised at the input resolution. Also, $H_1^{lm}$ and $H_2^{lm}$ are the low and middle frequency information of $H_1$ and $H_2$, respectively. The HR image is hallucinated in 3 steps:

1. **Global multimodal LR face synthesis**: Using the tensor representation and multilinear analysis which is a general extension to linear methods such as PCA and LDA, $\hat{L}_1$ and $\hat{L}_2$ are generated given $L_1$. This is done by ML estimation of $\hat{L}_1$ given $L_1$, and then using the multilinear analysis to generate LR images of the other modality ($\hat{L}_2$).

2. **Local patch-based face image hallucination**: Using a local patch-based multi-resolution tensor structure, the low and middle frequency correspondences of $\hat{L}_1$ and $\hat{L}_2$ are hallucinated by maximising $p(H_1^{lm}|\hat{L}_1)$ and $p(H_2^{lm}|\hat{L}_2)$ independently. The maximisers are approximated analytically using multilinear analysis for each patch in the images. The final hallucinated images ($H_1^{lm}$ and $H_2^{lm}$) are then a composition of the corresponding hallucinated patches.

3. **High-frequency residue recovery**: Using an MRF model and MAP estimation, each $H_i$ is inferred from the corresponding $H_i^{lm}$ by maximising $p(H_i|H_i^{lm}) = \prod_q p(H_i^{lm}|H_{i,q})p(H_i)$, where $q$ indicates the patch numbers. The difference between $H_i$ and $H_i^{lm}$ is in the high-frequency band which in turn depends on the middle and low frequency information. Hence, the likelihood term is defined using the Laplacian images ($\mathcal{L}$) of each patch in $H_i^{lm}$ and patches from the HR training data\(^2\). The patch $H_i^{(tr)}$ from the training HR data that has $\mathcal{L}(H_i^{(tr)})$ closest to $\mathcal{L}(H_i^{lm})$ is the most probable to be chosen as $H_{i,q}$ according to this likelihood term. Furthermore, by modelling the HR image as an MRF, the prior is defined as the product of the compatibility functions over all neighbouring patches: $p(H_i) = \prod_{H_{i,q},H_{i,\bar{q}}} \phi(H_{i,q},H_{i,\bar{q}})$ where $H_{i,q}$ and $H_{i,\bar{q}}$ are a neighbouring patch pair in a 4-neighbour system.

The novelty of this method is the ability to construct HR images of different modalities than what is present in the input LR image. However, this ability is still limited to the modalities which are available in the sample set. In fact, this is a general limitation of the example-based methods. Since these methods rely on their sample set as a source of information, they are naturally limited to the type of information it provides. Therefore, these methods are in general only able to reconstruct face images of modalities represented by the samples set, with very limited ability to generalise to new unseen modalities (poses, illuminations, expressions, etc.).

\(^2\)Note that the Laplacian image is used to represent middle-frequency information.
3.3 Direct HR-LR Matching

Most of the previously mentioned approaches for LR face recognition use super-resolution with the aim of reconstructing an HR image, with recognition only as an after-thought. In contrast, a rather new stream of methods has recently emerged in the literature with the direct aim of recognition, avoiding super-resolution as an initial pre-processing step.

The tensor-based framework of Jia and Gong ([56]) which was mentioned in the previous section, has been applied to simultaneous super-resolution and recognition ([55]) by directly computing a maximum likelihood identity parameter vector. However, that framework is not optimal in the sense of recognition, and the recognition comes merely as a by-product of the SR optimisation process. Hence, the framework was included in the previous section along with other two-step approaches.

Yeomans et al. [47] proposed an alternative approach which is aimed at simultaneous super-resolution and recognition (S2R2). In their approach, face features as they would be extracted for a face recognition algorithm (e.g., eigenfaces [103], Fisherfaces [12], etc.), are included in a super-resolution method as prior information. This approach simultaneously provides measures of fit of the super-resolution result, from both reconstruction and recognition perspectives. They approach the problem of face recognition by defining a distance metric $D(f_g, f_p)$ between the gallery and probe features, where $f_g$ is the vector of features extracted from the gallery image and $f_p$ is the feature vector extracted from the probe image. Using linear features, the feature vectors can be expressed as $f_\nu = F_h\nu$ where $\nu$ can be $g$ for gallery or $p$ for probe, $h$ is the HR image represented in vector form, and $F$ is the feature matrix. The problem of face verification is then thought of as comparing the distance $D(F_hg, F_hp)$ to a threshold in order to accept or reject a given claim while face identification can be thought of as finding the class among the gallery set with minimum distance from the given probe image. The distance metric can be written as:

$$D(f_g, f_p) = \|F_hg - F_hp\|^2$$  \hspace{1cm} (3.12)

In the absence of an HR probe image, the features, $F_hp$, cannot be calculated. Instead, the LR probe image, $l_p$, is used to produce an estimate, $\hat{h}_p$, of the HR image which will then be used to extract the face features: $\hat{f}_p = F\hat{h}_p$. Using Tikhonov regularization [114] as a simple super-resolution method, the HR estimate $\hat{h}_p$ is obtained by:

$$\hat{h}_p = \arg\min_h \left\{ \|Mh - l_p\|^2 + \mu^2\|Bh\|^2 \right\},$$  \hspace{1cm} (3.13)

where $M$ is an observation model to construct an LR image given an HR image, similar to that of Equation 3.2, $Bh$ is a vector of edges, and $\mu$ is a regularisation parameter.

The algorithm then proceeds to perform the super-resolution procedure by using Equations 3.12 and 3.13 jointly in a regularised fashion. Assume that for a given LR probe image, $l_p$, the $k^{th}$ class was claimed and we can compute or look up the gallery features
\( f_g^{(k)} = Fh_g^{(k)} \) using the gallery, where the superscript \( k \) indicates the claimed class. The main task of the S2R2 algorithm is then to find \( \hat{h}_p^{(k)} \) as:

\[
\hat{h}_p^{(k)} = \arg\min_h \left\{ \|Mh - l_p\|^2 + \mu^2\|Bh\|^2 + \tau^2\|f_g^{(k)} - Fh\|^2 \right\},
\]

where \( \tau \) is an additional regularisation parameter. Using the obtained \( \hat{h}_p^{(k)} \), the norms of the terms in Equation 3.14 are then calculated and stacked together to form a feature vector used for recognition. The first term of Equation 3.14 measures the fit between the observed low-resolution probe image and the low-resolution version of the resulting super-resolved image. The second term measures the smoothness of the super-resolution result, and the last term is a measure of the difference between ideal features for the claimed class and the features that would be produced by the super-resolution result. Note that, in a face identification scenario, such a feature vector needs to be computed for each gallery image in the gallery set separately. Hence, a major disadvantage of this framework is the high computational cost associated with this requirement. Yeomans et al. also extended their formulation to multiple resolutions and provided empirical results which suggested that in some cases using multiple resolutions simultaneously for matching can yield low error rates outperforming even those obtained with HR probe images (which in reality are not available). Another extension of this algorithm was presented in [48] where it was extended to multiple frames and multiple cameras.

A different approach which considers the recognition task directly and avoids SR as a pre-processing step was proposed by Li et al. [63]. This method uses coupled metric learning to find two transformations which map the LR and HR images into a new joint space where the new distance measure is more suitable for recognition. One transformation maps the LR probe images to a new space, where higher recognition performance can be achieved; the other maps the HR gallery images along with their class labels to the same space for better class-wise feature representation. The transformations are learnt using a set of labelled HR training images and their LR counterparts. Let us represent the set of mapped training LR images as \( \tilde{X} \), given as \( \tilde{X} = AX \) where each column of matrix \( X \) is a vector representation of a training LR image. The matrix \( A \) defines a linear mapping which maps these images into the new space. Consequently, each column of matrix \( \tilde{X} \) is a vector representing the corresponding LR training image in the new space. Also, let us denote by \( \tilde{Y} = F\hat{Y} \) the mapping of the HR training images together with their labels to the same space. Each column of matrix \( \tilde{Y} \) is a vector representation of an HR training image concatenated by a vector representing its class dependency. Thus, each column of \( \tilde{Y} \) embeds information about the corresponding training HR image as well as its label. The coupled transformations, \( A \) and \( F \), are then jointly determined by optimising an objective function defined as \( \|\tilde{X} - \tilde{Y}\| \), where \( \|\cdot\| \) denotes the Frobenius norm. The optimisation procedure is fairly efficient with closed form solutions. An extension of this work was introduced in [62] where a penalty weighting matrix is used in the objective function to maintain locality.

Another approach based on projecting the gallery and probe images into a new space was introduced by Biswas et al. in [17]. This method which was initially proposed for the case where both gallery and probe images are LR, is based on Multidimensional Scaling (MDS). A set of HR and LR training images are used in order to learn a
transformation which maps the LR images to a Euclidean space where the distance between the transformed samples approximates the best distance had both images been HR.

This method is extended to the case of HR gallery and LR probe images in [16] where two separate transformations are learnt for mapping feature vectors of the HR gallery image and the LR probe image to a new Euclidean space such that the interdistance between them closely approximates the distance had both images been of high resolution. More specifically, if the distance between the feature vectors of the \(i\)th and \(j\)th training HR image is denoted by \(d_{i,j}^h\), then the objective function to be minimised can be expressed as \(\sum_{i=1}^{N} \sum_{j=1}^{N} [D_{i,j} - d_{i,j}^h]^2\), where \(N\) is the total number of HR and LR training samples and \(D_{i,j}\) represents the distance between the \(i\)th HR training image and the \(j\)th LR training image after they are both mapped to the new space. In order to improve the matching performance, the above objective function is modified to include class information of the training data. This is realised by adding a class separability term which penalises larger distances between data points of the same class but does not affect data points of different classes. This approach is also extended to video in [15].

A similar concept was used in the recent work of Zhou et al. [113] where the HR and LR images are mapped to a common space. Inspired by the classic Linear Discriminant Analysis (LDA) [12], Zhou et al. proposed a Simultaneous Discriminant Analysis (SDA) framework in which two separate transformations are learned for mapping LR and HR images to a common subspace. The mapping is learned using a training set of HR and LR images, such that in the new subspace discrimination property is maximised. The parameters of the transformations are learnt by maximising an objective function. Similar to the conventional LDA, this objective function is defined as the ratio of the between-class and within-class scatters, after projection to the new space. The difference is that here two separate transformations are simultaneously learnt for LR and HR samples, respectively.

### 3.4 Model-Based Approaches

Alongside the view-based approaches to face recognition, a group of methods have been proposed in the literature which use a model of the human face for facial analysis applications such as face recognition. However, although many approaches have been proposed in the literature for model-based recognition of faces in high resolution, the application of such methods to low-resolution face recognition has been rather limited. In this section we first present the general model-based approach to face recognition and discuss the most popular models applied to this problem in high resolution. We then discuss how such methods have been adapted for the case of low-resolution facial analysis and recognition.

Many different face models have been used in the literature for various different applications. These include generic models describing the properties of an average face, deformable models which describe the properties of an arbitrary face and can fit to any given face, and even person-specific models which model a particular individual’s
face. The most widely used class of models in face recognition applications are the deformable models. Hence, we focus our attention mainly on this class. In the following, the word *model* is used to refer to a deformable model unless otherwise stated.

Generally speaking, in order to build a face model, a set of training samples are used to learn prior knowledge about the human face and build a model which can capture facial variations. This model is then used in an analysis step in order to describe a given facial image. An example of such models is the 3D Morphable Face Model (3DMM) reviewed in Chapter 2. The analysis step which involves fitting the model on an input image was reviewed in Section 2.4. The outcome of the fitting step is a set of model parameters which describe the input face. These parameters can then be used to identify the input face as discussed in Section 2.6.

The 3DMM is not the only face model used in facial research. Numerous different models have been proposed in the literature and applied to different problems involving the human face such as tracking, pose estimation, coding, and face recognition. Such models include statistical linear models capturing the shape and/or texture of the face (e.g. [27, 30]), non-linear models based on Multi-Layer Perceptrons [41, 59], physical and anatomical models [101] and possibly others.

We focus our discussion to those models that are most relevant to the subject of this thesis. More specifically, we focus on linear statistical models of the face appearance which model facial shape and/or texture using a set of sample images.

The *Active Shape Model* (ASM), proposed by Cootes *et al.* [27, 28], models the 2D shape of an object by a set of manually labelled landmarks where each labelled point represents a particular part of the object or its boundary. This model works by modelling how different points tend to move together. The training images are aligned by finding a global transform consisting of a rotation, translation, and scaling. Principle Componenets Analysis is then applied to the aligned shapes to learn a compact set of parameters which describe the shape of the object as observed by the variations within the sample set.

The ASM model only models the facial shape. To account for texture variation, this model was augmented by texture (grey-level) information [60, 61] to obtain a statistical appearance model using two complementary approaches. In the first approach, a flexible “shape-free” appearance model was built by warping each training image to have the same shape as the mean face and sampling grey-level values from the area within the face. A PCA transform was then applied to the vectors of grey-level samples to train a flexible “shape-free” grey-level model. In the second approach, a large number of local grey-level profiles were used, one for each landmark of the shape model. Each grey-level profile describes the texture variations in a region around a given landmark. By applying PCA to each of these profiles in a training set, a set of flexible grey-level models are learnt which model the local texture variation around each landmark. Thus, three flexible models are learnt for shape, grey-level profile, and shape-free grey-level. Collectively, these models are known as an *Active Appearance Model* (AAM). We refer to such a model as an *independent* AAM since it assumes that the shape and texture of a face are independent.

In contrast to the independent AAM, a *combined* AAM was proposed in [35, 37] which
3.4. Model-Based Approaches

considers the correlation between shape and texture of a given face. First, a shape model and a shape-free texture model are learnt using an approach similar to the above. Next, a combined vector is formed for each training sample by concatenating the shape and texture parameters into one vector. A further PCA is then applied to these combined vectors to obtain a compact set of parameters representing both shape and texture, simultaneously.

Different fitting algorithms have been proposed for such models. For instance, [35] and [61] use the grey level profile models to fit the shape of the model to a given face. The mean shape is projected to the image and iteratively updated to better fit the image evidence, subject to shape constraints represented by the shape model. At each step, a new proposed shape is obtained by searching the region around each landmark point for the best match to the corresponding local grey level model learnt during training. A similar approach was used in [27] for fitting the Active Shape Model. Hence, this approach is known as the ASM search [37]. In case of the AAM, after fitting the shape using the described ASM search, the grey level values of the input image are sampled within the face area and projected to the model space using the shape-free grey level model. The ASM search is an efficient method; however, the same authors state in [36] that this approach is not always robust since it does not use all the available information.

The most common approach to fitting an AAM on an image is by minimising the L2 norm of the error between the synthesised image and the input image warped back to the model coordinate frame. This norm can be expressed as:

$$\sum_{x \in s_0} [I(W(x; a)) - A(x; b)]^2$$  \hspace{1cm} (3.15)

where $x$ is the 2D coordinates of a point in model’s coordinate frame, $s_0$ is the mean shape, and $W(x; a)$ is a piece-wise affine transform defined by the model’s shape parameters, $a^3$. Also, $A(x; b)$ is the shape-free texture model defined in the coordinate frame of the mean shape and parametrised by the texture parameters, $b$.

According to Equation 3.15 for each point, $x$, in the coordinates of the model, the corresponding point in the coordinate frame of the input image is given by $W(x; a)$. This indicates warping image $I$ to the coordinate frame of the model. The sum is then taken over the points in this coordinate frame.

Optimising Equation 3.15 is a computationally expensive and time consuming task. In [29] and [30], Cootes et al. suggested an efficient solution by assuming that a linear relationship exists between the model parameter displacements and the residual error images. In other words, it is assumed that given the difference between the current model estimate and the input image, the necessary displacements to optimise the model parameters can be calculated in each iteration through a linear relationship. The authors proposed to learn this linear relationship during training and use it during the

\[\text{Note that the model is often augmented to include a global 2D transform as well as the linear shape variation parameters and the piece-wise affine warp parameters. See [73] for a brief discussion on two alternative approaches.}\]
fitting in order to calculate the displacements to model parameters efficiently. However, the assumption that such a relationship is linear does not always hold. Another efficient fitting algorithm which avoids making such assumptions was proposed in [73] by Matthews and Baker. Their method is based on an extension of the Inverse Compositional Image Alignment (ICIA) [11] algorithm.

In all the above models and associated fitting algorithms, the model and the input image are assumed to be of approximately similar resolutions. Note that in order to calculate the difference between a given image and the synthesised image, Equation 3.15 describes first warping the input image to the coordinate frame of the model, then comparing it against the face synthesised by the model. Naturally, if the resolution of the input image is lower than the model’s resolution, this warping would include some sort of interpolation which can introduce an unwanted bias in the fitting procedure.

In [34] Dedeoglu and Kanade argue that the image registration problem where one image is warped to another for comparison, is not a symmetric problem and that there exists a natural choice for the direction of the warp. In particular, by analysing the image registration criterion, they concluded that in order to obtain an unbiased estimate of the warp between two images one should chose the direction of warp such that, after necessary blurring, it scales one image down onto the other. In other words, one must start with the higher resolution image, blur it appropriately, then warp it onto the low-resolution one. This is in contrast to fitting an AAM to a low-resolution image using Equation 3.15 which warps the image (low-resolution) onto the model (high-resolution).

Accordingly, Dedeoglu et al. proposed a resolution-aware fitting algorithm for fitting a 2D Active Appearance Model to LR face images [33]. Their resolution-aware formulation follows the above-mentioned natural choice and instead of warping the input image onto the model, it uses an imaging model which blurs the model’s appearance and warps it onto the input LR image. The objective function of this algorithm takes the form:

$$\sum_{m \in I} [I(m) - B(m; A(W(a); b))]^2$$  \hspace{1cm} (3.16)

where $m$ is the 2D coordinates of a point in the coordinate frame of the LR input image, and $B(\cdot)$ is an imaging model which simulates an LR image of the face. Note that in this formulation the sum is taken over the LR pixels whereas in the conventional fitting criterion of Equation 3.15 the objective function was summed over pixels in the model’s coordinates.

The algorithm presented in [33] avoids biased estimates by avoiding the interpolation discussed above. This algorithm was only applied to face tracking where each experiment was initialised with the fitting results on high-resolution images and in most experiments a person-specific AAM was used. Whether the model estimates acquired using this method are accurate enough for tasks such as person identification under unconstrained scenarios is not evaluated and remains unclear.

In [68], Liu et al. proposed an alternative approach to address the problem of fitting an AAM on a low-resolution image. Their approach uses a conventional fitting criterion
Model-Based Approaches

(Equation 3.15), but uses a multi-resolution AAM where the effective resolution of the model is chosen based on the resolution of the input, hence avoiding significant interpolation of the input image during fitting. The training data is first down-sampled to lower resolutions at different scales. An AAM is then trained at each resolution. The landmarks at lower resolutions are obtained by scaling the landmarks from the HR images. Thus, the mean shapes of the multi-resolution AAM differ only by a scaling factor while the shape basis vectors are exactly the same across different resolutions. During fitting, an appropriate model resolution is chosen based on the resolution of the input image. The authors compared the fitting convergence of models at different resolutions and concluded that the best performance is obtained when the resolution of the model is only slightly higher than the input image. Furthermore, a face tracking experiment using a person-specific AAM was performed and the results showed that when fitting to an LR image, using an AAM with a resolution close to the input yields better performance compared to using a high-resolution AAM. The authors did not apply their method to face recognition in [68]. Hence, it is not clear whether the parameters estimates obtained with this method are robust enough for recognition. However, the same authors used this approach in a model-assisted framework for super-resolving facial texture as a pre-processing step for face recognition [108]. An image formation model similar to Equation 3.1 was used where the registration was performed by fitting a multi-resolution AAM on the LR inputs. The authors then used an approach similar to [40] for super-resolution where the super-resolution criterion function is an L1 norm of the difference between the model of the observations and the actual observations, plus an L1-based regularisation term. The authors reported improved recognition results using a commercial face recognition engine (FaceIt SDK ver. 6.1, Identix Inc.). However, the probe set used for their experiments contained images of only 3 identities captured by the authors. Furthermore, although the results were categorised by eye-to-eye distance of the probe image, the relative resolution of the gallery set - and therefore the effective magnification rate - is not clear and the recognition rates reported for the lowest probe resolutions are not significant. More specifically, for eye-to-eye distance equal to 19 and 17 pixels, the rank-1 identification rate was 33% and 13%, respectively. For these reasons, the results reported in [108] cannot be considered extensive enough to confirm that the method can significantly improve face recognition in low resolutions.

A similar concept of using a multi-resolution model was used by Kang and Buyan in [58] for 3D Morphable Models. They built a multi-resolution model by sub-sampling a high-resolution face model to obtain the lower resolution models. The fitting procedure in this framework starts by fitting the model with the lowest resolution, the optimal parameters of each level are then used to initialise the fitting process for the next (higher resolution) level. The main purpose of that work was to show that fitting efficiency can be improved by employing a multi-resolution approach. Hence, no comparison was made as to how a low-resolution model could be used to improve fitting accuracy on low-resolution input images. This was investigated in a recent work by Hu et al. [49]. They compared the fitting result of three different models with different resolutions (number of vertices) on low-resolution and high-resolution images and verified that when the input is LR, fitting a model with fewer vertices yields more accurate results than fitting a high resolution model.
In addition to fitting a model directly to LR input images, an alternative approach to benefit from an explicit face model in LR face analysis is to use the model in order to assist face super-resolution. An example of such an approach was already discussed above. Another example is the work of Yu et al. [111] where a 3D model is used to assist facial texture super-resolution from a video sequence. A generic 3D face model is used to track the face in the video. Pose and illumination information are estimated from the incoming frame using the 3D model. This information is then used to super-resolve the facial texture using Iterative Back-Projection (IBP). Finally, the super-resolved texture is fed back to the tracking module to improve the estimation of pose and illumination. This process is continuously repeated to refine the facial texture as new frames of the video become available. Experimental evaluation of this method was only limited to one synthetic and one real video sequence where the authors demonstrated that their approach is able to integrate information from multiple frames in order to reconstruct the facial texture at a higher resolution. This approach was used in [110] for face recognition in LR video. Both gallery and probe sets consist of LR video sequences. The results show that this method can improve face recognition performance when a LR video sequence is available, especially when information from many frames is integrated to construct the SR image.

3.5 Discussion and conclusions

This chapter reviewed face recognition using low-resolution probe images. We categorised the different approaches to this problem into four main categories and discussed various methods within each category. However, without making assumptions about availability of sample training images with various poses and illuminations, the methods discussed in this chapter are not suitable for the purpose of this thesis which is pose- and illumination-independent face recognition in low-resolution still images.

We discussed that down-sampling the gallery set to match the resolution of the probe image (first category of methods) is not an optimal approach since considerable amounts of discriminatory information are lost in the low-resolution gallery and probe images. Furthermore, it is impractical to generate a separate gallery set for each input.

The second category of methods, uses super-resolution as a pre-processing step to enhance the probe image prior to recognition. The main source of information for super-resolution algorithms is the reconstruction constraint which states that the super-resolution image, when appropriately warped and down-sampled to mimic the image formation process, should yield the low-resolution input image(s). While reconstruction-based super-resolution methods provide a means to enhance any image, including a facial image with any pose or illumination, their potential application to LR face recognition is limited in a number of ways. Firstly, in order to reconstruct an HR image, these methods require multiple LR images which may not be available in a practical face recognition scenario. Secondly, most of these methods heavily rely on sub-pixel accurate registration of the LR inputs, which itself can prove to be a very challenging task. The third and possibly most important limitation of such methods as discussed in Section 3.2.2, is the fact that there is a theoretical limit to how well the reconstruction-based super-resolution methods can perform. The limit effectively means that, even
if multiple frames are available and accurately registered, the typical magnification ratios achieved by such methods are generally insufficient for face recognition in low resolution.

The limits of reconstruction-based super-resolution can be overcome by example-based methods. However, these methods are prone to limitations imposed by the finite sample set. Example-based methods learn or model relationships between the LR and HR faces and utilize it as an additional source of information to enhance the super-resolved image, but the learnt information is limited to poses and illuminations available in the sample set. A human face can have completely different appearances when viewed from viewpoints (poses) or under illumination conditions which are different to those previously seen in the sample set. Moreover, in the case of a highly deformable object such as the human face, the object’s deformation (e.g., facial expression) can further complicate the task of modelling its appearance using a finite set of 2D images. Naturally, example-based methods which mainly rely on their sample set to model (almost) all possible appearances of the human face are bound by their finite sample set and their performance will be highly degraded, if not completely compromised, when reconstructing faces of previously unseen poses or illumination conditions. Figure 3.4 compares super-resolution results using the example-based approach of Baker and Kanade [10] for two input images. A sample set consisting of frontal faces was used for generating these results. It is evident that the approach yields good results (Figure 3.4(c)) for the frontal LR input which matches the pose of the sample set. However, when the input LR image has a previously unseen pose, the result is unacceptable (Figure 3.4(d)). Notice that the algorithm has tried to enhance the non-frontal image by adding high-frequency information which is extracted from sample frontal faces which is obviously not suitable for this image.

A similar argument applies to subspace learning methods which fall in the third category. Here, the training set used for learning the necessary transformations effectively determines the limits for applying such transformations. Hence, these methods cannot be directly applied for recognition of faces with a previously unseen pose or illumination.

The fourth category of methods is the model-based approach which utilises an explicit face model built from exemplar faces. To the best of our knowledge, among the model-based methods of Section 3.4, only the AAM model has, to a limited extent, been applied to the problem of low-resolution face recognition from still images. Being a 2D model, the AAM is naturally limited in terms of generalisation to new poses and illuminations. Although multi-modal Active Appearance models which model different poses and illuminations have also been proposed in the literature, these approaches also require examples of these different modalities to build the face model. Hence, these approaches are also limited by the finite sample set used for building the model.

In conclusion, any method that only relies on information learnt from a finite set of 2D facial images in order to enhance, or model a LR face for the purpose of recognition is likely to have limitations in using such learnt information to recognise previously unseen poses and illuminations. This is because intrinsic and extrinsic information are combined together in a 2D image. Hence, the learnt information is affected not only by the intrinsic factors which are valuable for recognition, but also by extrinsic factors which are not related to the subject’s identity and can have a negative effect on the
Figure 3.4: Face super-resolution using the approach of Baker and Kanade [10]. (a) and (b) show a frontal and a non-frontal LR face, respectively, both enlarged by a factor of 8 using bilinear interpolation. (c) and (d) show the results of super-resolving the LR images using a sample set consisting of frontal faces. While super-resolution of the frontal input yields acceptable results, it fails for the non-frontal input which does not match the pose of the sample set.
3.5. Discussion and conclusions

recognition performance.

As discussed in the previous chapter, a powerful tool which can overcome such limitations is the 3D Morphable Face Model (3DMM) which uses exemplar 3D facial scans to model class-specific knowledge about human faces independently of extrinsic factors such as pose or illumination. Thus, one can separate different factors affecting the face’s appearance in a given image and handle each of them appropriately. The 3DMM can then be used as a generative model in order to synthesise previously unseen appearances of the human face, thereby providing a means to generalise limited prior information obtained from a finite set of examples. Hence, one could expect to overcome the problem of low-resolution face recognition in an unconstrained scenario, i.e. independent of the facial pose and scene illumination, by appropriately fitting a 3DMM model to an LR face. The information provided by this model fitting can then be used for analysis of the LR face. This is the main motivation of the work presented in this thesis. We will address the problem of fitting a 3DMM to low-resolution images allowing LR face recognition, as well as HR face rendering, to be performed using only a low-resolution input. Our approach infers HR information in the form of a 3D high-resolution model from a single 2D low-resolution image of virtually any pose or illumination; a goal not achievable using any of the methods reviewed in this chapter. We will demonstrate that such information can then be used for face recognition. We will also show how to further enhance such information by presenting a novel framework for pose- and illumination-independent facial texture super-resolution. This framework which is presented in Chapter 6 is aided by the aforementioned 3D information and allows for super-resolution of the facial texture of a given LR face, independent of the pose and illumination conditions. We will also demonstrate how this super-resolved facial texture can be used for rendering an HR face, which can in turn be used in an alternative method of LR face recognition.
Chapter 4

Resolution-Aware Fitting of 3D Morphable Models on Low-Resolution Images

4.1 Introduction

In Chapter 2 we reviewed the 3DMM as a tool for unconstrained face analysis which can overcome many challenges that normally impact the performance of a 2D face recognition system. We mentioned that analysis of 2D facial images is performed using a process known as model fitting and we described the main approaches to model fitting under the implicit assumption that the resolution of the input image is high.

In this chapter we consider the case where the input image has low resolution and we show that, although the previously mentioned assumption was not explicitly made by the conventional model fitting approaches of Chapter 2, those approaches would fail if the assumption does not hold. By critically analysing the criteria commonly used by the main fitting algorithms and comparing them with a continuous image formation model, we show that these criteria are only valid if the resolution of the input image is high.

Inspired by the Super-resolution literature presented in Chapter 3, we propose to improve 3DMM fitting on LR images by formulating an image formation model to describe the process of LR image formation given the 3D model and using this imaging model to formulate a pixel colour criterion for LR input images.

Furthermore, we argue that the edge cost function used in the MFF framework becomes highly biased when using LR input images. We propose to address this by constructing a smooth edge cost surface via a multi-resolution approach. We use this cost surface to formulate an additional criterion for 3DMM fitting to LR images.

Finally, we evaluate our proposed framework for LR-specific 3DMM fitting. Experimental results show that our algorithm significantly improves fitting on LR images and yields similar parameters to those that would have been obtained if the input image had a higher resolution.
4.2 The Resolution Problem

A digital image is formed by interaction of light with an array of sensor elements which are responsive to visible light energy. A typical sensor for digital cameras is a CCD. Light reflected by the objects inside a scene passes through the camera optics and reaches the CCD. The light energy (irradiance) that reaches each CCD element is then transformed into a voltage which is in turn digitised to yield a value representing the colour at the position corresponding to the given CCD element. The output voltage of each CCD element is proportional to the integral of the irradiance projected onto its photosensitive surface.

The above process can be modeled as the convolution of the continuous irradiance light field that reaches the image plane by the Point Spread Function (PSF) of the camera. Let \( m = (m, n) \in \mathbb{Z}^2 \) be the 2D pixel indices on the image plane of \( I \). The continuous image formation model can be expressed as [79]:

\[
I(m) = (E \ast PSF)(u) = \int_u E(u) \cdot PSF(u - m) du, \tag{4.1}
\]

where \( E(.) \) is the continuous irradiance light-field that would reach the image plane, \( PSF(.) \) is the point spread function of the camera, and the integral is taken over \( u = (u, v) \in \mathbb{R}^2 \) which is the vector representing the continuous pixel coordinates on the image plane. Note that we have assumed that the 2D indices of a pixel, \( m \), correspond to the 2D coordinates of its centre.

PSF can further be decomposed into two components:

\[
PSF(u) = (w \ast a)(u), \tag{4.2}
\]

where \( w \) models the optical blur and \( a \) models the spatial integration performed by the CCD sensor elements. In the simplest case, the optical blur can be neglected (\( w(u) = \delta(u) \)). Furthermore, we assume that the CCD elements are uniformly sensitive to light. Hence, \( a(u) \) takes the form:

\[
a(u) = \begin{cases} 
\frac{1}{A} & \text{if } u \in bin(m) \\
0 & \text{otherwise}
\end{cases} \tag{4.3}
\]

where the condition \( u \in bin(m) \) means that \( u \) is a point lying in the photosensitive region of pixel \( m \), and \( A \) is the photosensitive area of each pixel.

In the context of synthesising a 2D image from the 3DMM, one can assume that the irradiance light field that reaches the virtual camera’s imaging plane is the illuminated and colour-transformed texture, \( t^C \), of the model. This irradiance is defined in the 3D object-centred coordinates, \( x \), of the model as opposed to \( E(u) \) in Equation 4.1, which is defined in the continuous 2D image coordinates. However, assuming \( t^C(x) \) is continuous in \( x \), and the inverse shape projection, \( p^{-1}(u) \), is continuous in \( u^1 \), the continuous irradiance light-field that reaches the image plane can be expressed as:

\[\text{Recall from Section 2.3 that continuity can be provided by interpolation using the triangle list.}\]
4.2. The Resolution Problem

\[ E^{\text{model}}(u) = t^C(x) \circ p^{-1}(u) \] (4.4)

In this equation the symbol \( \circ \) represents composition as defined by Equation 2.12 and \( x = p^{-1}(u) = (x, y, z) \in \mathbb{R}^3 \) is the 3D location of the point on the model surface which projects to the image location \( u \).

Recall that the fitting algorithms discussed in Chapter 2 use an analysis-by-synthesis approach where the synthesised image, \( I^{\text{model}} \), is given by an image formation model which describes the process of rendering a 2D image, given the 3DMM parameters and the rendering parameters. This image formation model was given in Equation 2.22. For ease of reference, we include that equation here again with simplified notation. The image value at a given pixel, \( m \), is given by:

\[ I^{\text{model}}(m) = t^C(x) \circ p^{-1}(m) \] (4.5)

where we have made all the dependencies on model parameters implicit for simplicity.

Comparing Equations 4.5 and 4.4, it is evident that the synthesised image used by the conventional fitting algorithms (Equation 4.5) is actually the irradiance, \( E^{\text{model}}(u) \), sampled at the pixel centre locations of the image plane. Such algorithms neglect the effect of the convolution with the camera point spread function, effectively assuming \( PSF(u) = \delta(u) \).

From Equation 4.3, it is apparent that such an assumption can only be justified if the size of the CCD elements can be assumed to be infinitesimally small (ie. \( A \to 0 \)):

\[ \lim_{A \to 0} a(u) = \begin{cases} \delta(u) & \text{if } u \in \text{bin}(m) \\ 0 & \text{o.w.} \end{cases} \] (4.6)

In the case of a high resolution image (dense sampling grid), the pixels can be assumed to have an infinitesimally small size (\( A \to 0 \)); thus, the above assumption of \( PSF(u) = \delta(u) \) can be justified. However, as the image resolution decreases, the effect of spatial integration becomes increasingly significant making the imaging model of Equation 4.5 and the corresponding fitting criterion (Equation 2.24) increasingly sub-optimal. Thus, for a low-resolution input, where the pixel size cannot be assumed infinitesimally small, the image formation model must consider the spatial integration in order to synthesise a more realistic image. In Section 4.3 we will derive such an image formation model, suitable to model the LR imaging process.

Figure 4.1 illustrates the cases of HR and LR image synthesis. The model triangles are projected to the image plane to synthesise the irradiance light field that reaches this plane. This field is then sampled by the image pixels. In the case of synthesising an HR image, where the image pixels are small relative to the projected triangles, point sampling at the location of a pixel centre yields a good estimate for the value of that pixel. However, in the case of LR image synthesis the image pixels are much larger than the projected triangles and a large number of triangles project to the same image pixel.
Chapter 4. Resolution-Aware Fitting of 3DMM on LR Images

Figure 4.1: Image synthesis using the conventional image formation model. Top: HR image synthesis, note that the size of projected model triangles is comparable to the size of the image pixels. Bottom: LR image synthesis. Image pixels are much larger than the projected triangles. As a result, multiple triangles project to each image pixel.

In this case, back-projecting a single point to the model does not yield an accurate estimate of the pixel value, which in fact is the integral of texture values of all triangles projected to the given pixel.

4.3 A Low-Resolution Image Formation Model

We derive a model to synthesise a low-resolution image given the shape and texture parameters of the 3D face model, as well as the projection and illumination parameters. As discussed previously, such a model must consider the effects of spatial integration by the CCD elements. In order to consider the effects of spatial integration in the imaging model, we need to compute the continuous irradiance field, $E(u)$, over the whole image plane as opposed to the conventional imaging models which only sample this irradiance field at the location of the pixel centres.

One could expect that it is sufficient to replace the irradiance in Equation 4.1 with $E_{\text{model}}(u)$, as given by Equation 4.4, and take the integral over the image plane. However, taking a closer look reveals that this is not a feasible task since a) the illuminated model texture, $t^*(x)$, is not continuous over $x^2$; and b) as mentioned in Chapter 2, the inverse projection $(p^{-1}(u))$ cannot be expressed analytically.

In the following, we present a LR image formation model which estimates the continuous irradiance field over the whole image plane, considers the spatial integration

\footnote{Although continuity can be achieved through interpolation using the triangle list, the computational cost of this task is very high.}
over pixels, and is simple enough to be practical. We follow a similar path to the one presented in Section 2.3 for the case of high-resolution image synthesis, with modification and simplifications suitable for the LR case in order to satisfy these requirements. Considering that our aim is to render a low-resolution image, we first present a simple modification to the way texture is modelled in the 3DMM which greatly reduces the complexity involved in calculating the continuous irradiance field. We then proceed to model the image formation process.

4.3.1 Centroid-based Linear Texture Model

For the purpose of synthesising a low-resolution model, the variations of texture within a polygon of the model are negligible. Therefore, we assign a single texture value to each triangle defined as the average texture of its vertices. In other words, assuming that the texture of each triangle is constant and is equal to the texture value of the triangle’s centroid, we re-sample the texture model and compute a new linear model for the texture, based on the triangle centroids.

Let us denote the vertex indices of the three corners of the $k^{th}$ triangle by $s_1$, $s_2$, $s_3$, and their RGB texture values as given by the vertex-based linear texture model (Equation 2.8) by $t_{s_1}$, $t_{s_2}$, and $t_{s_3}$:

$$t_{s_1} = \bar{t}_{s_1} + \sum_{i=1}^{D_T} \beta_i t_{e_i; s_1}$$
$$t_{s_2} = \bar{t}_{s_2} + \sum_{i=1}^{D_T} \beta_i t_{e_i; s_2}$$
$$t_{s_3} = \bar{t}_{s_3} + \sum_{i=1}^{D_T} \beta_i t_{e_i; s_3}$$

(4.7)

where $\bar{t}_j$ is the average texture of vertex $j$, and $t_{e_i; j}$ is a subvector of the $i^{th}$ eigenvector ($T^e_i$) which corresponds to the $j^{th}$ vertex. The RGB texture value, $t_k$, assigned to the $k^{th}$ triangle in our re-sampled centroid-based model is then given as:

$$\hat{t}_k = \frac{1}{3} (t_{s_1} + t_{s_2} + t_{s_3}) = \hat{t}^{ave}_k + \sum_{i=1}^{D_T} \beta_i t_{e_i; k}$$

(4.8)

where we have used a hat symbol in our notation for the centroid-based model in order to distinguish it from the vertex-based model. In the above equation, $\hat{t}^{ave}_k$ is the average texture of the $k^{th}$ centroid over the training data:

$$\hat{t}^{ave}_k = \frac{1}{3} (\bar{t}_{s_1} + \bar{t}_{s_2} + \bar{t}_{s_3}),$$

(4.9)
and \( \hat{t}_{i;k} \) is the \( i \)th basis vector of the centroid-based linear texture model, corresponding to the \( k \)th triangle:

\[
\hat{t}_{i;k} = \frac{1}{3} \left( t_{i;s_1}^e + t_{i;s_2}^e + t_{i;s_3}^e \right)
\]  

(4.10)

Given \( \hat{t}_{k}^{ave} \) and \( \hat{t}_{i;k} \) for all triangles, we can define the average and basis vectors of the centroid-based texture model as:

\[
\hat{T}^{ave} = \left[ (\hat{t}_{1}^{ave})^T, (\hat{t}_{2}^{ave})^T, \ldots, (\hat{t}_{N_t}^{ave})^T \right]^T
\]

\[
\hat{T}^e_i = \left[ (\hat{t}_{i;1}^e)^T, (\hat{t}_{i;2}^e)^T, \ldots, (\hat{t}_{i;N_t}^e)^T \right]^T
\]  

(4.11)

where \( N_t \) is the number of triangles in the model. Finally, the texture values for the centroids of any given face are given as a linear combination of the above vectors:

\[
\hat{T} = \hat{T}^{ave} + \sum_{i=1}^{D_T} \beta_i \hat{T}^e_i
\]  

(4.12)

Note that Equation 4.12 defines a vector space which can be considered as a dual to the original vertex-based vector space of textures. The interesting property of this dual space is that the same mixture coefficients, \( \beta_i \), used in the original vertex-based space are used to obtain the centroid-based textures. Hence, we do not need to calculate new statistics in order to build this dual space. In fact, this space is only built through topological relationships with the original vertex-based vector space. Since the same mixture coefficients of the vertex-based model are used, the prior distribution defined by Equation 2.9 can still be used.

Using the new re-sampled texture model, the continuous texture can now be defined for every point on the model’s surface in the object-centred coordinates:

\[
\hat{t}(x) = \hat{t}_k, \quad \text{for all } x \in \text{tri}^o(k)
\]  

(4.13)

where \( \text{tri}^o(k) \) denotes the \( k \)th triangle in the object-centred coordinates. Note that the model’s texture, as described by Equation 4.13, is piece-wise constant.

### 4.3.2 Low-Resolution Image Synthesis

Using the centroid-based texture model described above, and considering that our aim is to render a low-resolution image, we now present our low-resolution imaging model. Similar to the HR case of Chapter 2, we start by presenting necessary functions for shape projection and texture illumination.
4.3. A Low-Resolution Image Formation Model

Shape Projection

Recall from Section 2.3 that once the object-centred coordinates of the vertices are computed using the linear shape model (Equation 2.8), the 3D object-centred coordinates, \( \mathbf{x} \), of each point are transformed to world coordinates, \( \mathbf{w} \), using a 3D rigid transform and the 3D world coordinates of the point are subsequently projected to the relative 2D image plane location using a perspective projection.

For the purpose of synthesising a low-resolution image, one can assume that the size of the object (polygon) is small relative to the distance between the object and the camera. Hence, we replace the perspective projection with a weak-perspective projection transform. Thus, instead of Equation 2.11, we use the following equations for projecting a 3D point from the world coordinates to the image plane:

\[
\begin{align*}
    u &= u_o + f \frac{w_{x,i}}{w_z}, \\
    v &= v_o - f \frac{w_{y,i}}{w_z}
\end{align*}
\]  

(4.14)

where \( u_o = (u_o, v_o) \) is the position of the optical axis on the image plane, \( f \) is the focal length of the virtual camera as before, and \( \bar{w}_z \) is the average depth (world coordinates) of the vertices of the triangle in which the given point lies.

We denote by \( \mathbf{p}(\mathbf{x}; \rho) \), the vector-valued function which projects a point \( \mathbf{x} = (x, y, z) \) from the 3D object-centred coordinates to the point \( \mathbf{u} = (u, v) \) on the 2D image plane, using the weak-perspective projection. The inverse, \( \mathbf{p}^{-1}(\mathbf{x}; \alpha, \rho) \), of this mapping can be defined in a similar way as \( \mathbf{p}^{-1}(\mathbf{x}; \alpha, \rho) \) was defined; i.e. using the triangle list.

Texture Illumination and Colour Transform

We use the Blinn-Phong reflectance model [21] to model the illumination of the continuous 3DMM texture, \( \mathbf{t}(\mathbf{x}) \), defined by the centroid-based linear texture model. Assuming that the 3DMM texture is illuminated with an ambient light and one directional light source, the continuous illuminated texture is given by the Blinn-Phong reflectance model as:

\[
\mathbf{t}'(\mathbf{x}) = \begin{pmatrix}
    L^a_r & 0 & 0 \\
    0 & L^a_g & 0 \\
    0 & 0 & L^a_b
\end{pmatrix} \cdot \mathbf{t}(\mathbf{x}) + \begin{pmatrix}
    L^d_r & 0 & 0 \\
    0 & L^d_g & 0 \\
    0 & 0 & L^d_b
\end{pmatrix} \cdot (\langle \mathbf{n}, \mathbf{d} \rangle \cdot \mathbf{t}(\mathbf{x}) + k_s \cdot \langle \mathbf{n}, \mathbf{h} \rangle \cdot \mathbf{1}_{3 \times 1})
\]  

(4.15)

where the vectors \( \mathbf{n} \) and \( \mathbf{v} \) are the surface normal and light direction vectors at point \( \mathbf{x} \), respectively. Also the vector \( \mathbf{h} \) is the halfway vector between light direction vector, \( \mathbf{d} \), and the viewing direction, \( \mathbf{v} \):

\[
\mathbf{h} = \frac{\mathbf{d} + \mathbf{v}}{\| \mathbf{d} + \mathbf{v} \|}
\]  

(4.16)
Chapter 4. Resolution-Aware Fitting of 3DMM on LR Images

Applying the colour transform of Section 2.3, the final texture to be imaged by the camera is given as:

\[
\hat{t}^C(x) = G\hat{t}^I(x) + o,
\]

where the matrix \(G\) and vector \(o\) are given in Equation 2.21.

We represent the above texture illumination and colour correction procedure by the vector-valued function \(\hat{t}^C(x; \beta, \gamma, \alpha, \tau)\).

### 4.3.3 The Low-Resolution Image Formation Model

In this section we present our image formation model which describes the process of rendering an LR image, given the 3DMM shape and texture parameters, as well as the set of projection and illumination parameters as defined in Chapter 2. We denote by \(\mu = \{\alpha, \beta\}\), the set of all model parameters while the set of rendering parameters consisting of projection and illumination parameters is denoted by \(\rho = \{\tau, \gamma\}\).

We start from the continuous image formation model of Equation 4.1. The continuous irradiance light field that reaches the image plane, \(E_{\text{model}}(u)\), is in fact the illuminated and colour transformed model texture, projected onto the image plane. Thus, we can compute this irradiance as:

\[
E_{\text{model}}(u) = \hat{t}^C(x; \mu, \rho) \circ \hat{p}^{-1}(u; \rho)
\]

Note that the difference between this equation and Equation 4.4 is that here the texture, \(\hat{t}^C(x; \mu, \rho)\), is the continuous centroid-based model texture; hence, there is no need for interpolation through the triangle list.

From Equations 4.1 and 4.18, the synthesised image can be expressed as:

\[
I_{\text{model}}(m) = \frac{1}{A} \int_{\hat{p}(x; \rho) \in \text{bin}(m)} |J(\hat{p})| \ \hat{t}^C(x; \mu, \rho) \circ \hat{p}^{-1}(u; \rho) \ d\hat{x}
\]

where we have assumed that \(PSF(u) \simeq a(u)\), and \(a(u)\) is defined by Equation 4.3. In the above equation, the integral is taken over \(u \in \text{bin}(m)\), which simply means that the integral is taken over the photosensitive area of pixel \(m\).

Considering that \(u = \hat{p}(x; \rho)\), we change the variable over which the integral is taken:

\[
I_{\text{model}}(m) = \frac{1}{A} \int_{\hat{p}(x; \rho) \in \text{bin}(m)} |J(\hat{p})| \ \hat{t}^C(x; \mu, \rho) \ d\hat{x}
\]
where $|J(\hat{p})|$ is the determinant of the Jacobian of the vector-valued function $\hat{p}(x; \rho)$. Note that the integral is now taken over the area on the 3D face surface which would project to pixel $m$ under the projection $\hat{p}(x; \rho)$.

Recall that the continuous texture, $\hat{t}(x; \mu, \rho)$, of our centroid-based model is piece-wise constant (Equation 4.13). Hence, the integral in Equation 4.20 can be expressed as:

$$I_{model}(m) = \frac{1}{A} \sum_{k \in K(m)} W(k, m) \hat{t}_{k}^C \quad (4.21)$$

where $\hat{t}_{k}^C$ represents the texture of the $k^{th}$ triangle ($\hat{t}_k$) after being illuminated and colour transformed, and $K(m) = \{ k \mid \hat{p}^{-1}(u; \rho) \in tri_o(k), u \in bin(m) \}$ is the set of triangles which, after projection to the image plane, overlap with pixel $m$. Also, $W(k, m)$ is

$$W(k, m) = \int_{\hat{p}(x; \rho) \in bin(m)} |J(\hat{p})| \, dx \quad (4.22)$$

It can be shown that the value of $W(k, m)$ given by the above equation is in fact the area of overlap, on the image plane, between pixel $m$ and the $k^{th}$ triangle after it is projected to the image plane.

Equation 4.21 defines an imaging model describing the image formation process given the 3DMM and a set of rendering parameters. Note that unlike the image formation models used in conventional fitting algorithms (Equation 2.22), our formulation takes into account the point spread function of the camera. More specifically, the spatial integration over each CCD element is modelled in our LR imaging model. We use this model to formulate a suitable criterion for fitting a 3D morphable model to LR images.

The Image formation model of Equation 4.21 is comparable to the imaging model commonly used in the super-resolution literature, given in Equation 3.3 (excluding the additive noise term which was included in Equation 3.3). Both of these equations describe the value of each LR pixel as a weighted sum over texture values of a set of HR spatial elements which overlap with the LR pixel at hand. In the case of the SR observation model, the spatial elements are the pixels of the sought HR image while in the case of our 3DMM image formation model, the spatial elements are the triangles of the 3DMM, projected to the image plane. In both cases the weights of the weighted sum depend on the amount of overlap between each HR spatial element and the LR pixel at hand.

### 4.3.4 The Reference Resolution

Spatial resolution is a broad term which roughly refers to the smallest discernible detail that can be visually recovered from an image. In this sense, there is no unified definition or measure for the spatial resolution of an image. In digital images, spatial resolution is usually expressed in terms of the number of pixels per unit length or area, e.g. pixels
per inch (ppi). However, this is merely an upper bound for the resolution since the image could, and almost certainly will, contain less information than what is possible to represent by all of its pixels. If the amount of high frequency detail in an image is considerably less than the maximum amount of detail representable by its pixel per area count, the image would appear overly smooth (blurred), despite having a high nominal pixel count per area.

Following from the above discussion, defining an exact measure for the resolution of an image is rather infeasible, and probably unnecessary, for most practical applications. However, considering the specific application at hand, one may attempt to define a measure in order to, at least conceptually, quantify the resolution of an image. In the low-resolution face analysis literature, it is common to express the resolution of a facial image in terms of the number of pixels between the eye centres, a measure known as the eye-to-eye distance. This is also merely a measure of pixels per length. Thus, it only defines an upper bound for the image resolution. Furthermore, it is not a consistent measure as it can vary with the subject’s pose and physical interocular distance. However, expressing the resolution of an image in terms of the eye-to-eye distance is a practically useful way of expressing the spatial resolution of a facial image under the implicit assumption that the image is not perceived blurry by the viewer.

In addition to the eye-to-eye distance, we define and use another convention in order to categorise facial images into low- or high-resolution in the context of our particular application: 3DMM fitting. We define a reference resolution as the conceptual threshold between high and low resolutions.

Considering the continuous image formation model and our analysis in Section 4.2, the reference resolution is the lowest resolution for which the spatial integration can be ignored and the colour value of a pixel, \( I^{model}(m) \), is determined by a point sampling at its centre. Ideally, the reference resolution should be defined as the resolution where there is a one-to-one correspondence between the spatial elements of constant texture, i.e. between the triangles of the model and the pixels of the image. However, for obvious reasons, this is not feasible. A less constraining definition for the reference resolution can be given as the lowest resolution at which a point sampling at the centre of a pixel yields a good estimate for the pixel value, or equivalently, the lowest resolution where a single triangle’s illuminated texture, \( t^C_k \), is a good estimate for the final texture value, \( I^{model}(m) \), of the pixel to which it projects. This is possible if the variance of the samples, \( \{t^C_k \mid k \in K(m) \} \), is low. Hence, a practical way of finding the reference resolution would be to find the variances of texture values of the triangles which fall within each pixel. If these variances are lower than a certain threshold, then a single triangle would yield a good estimate for the pixel value and the image can be considered HR. High variance, on the other hand, would mean that a single triangle cannot represent the pixel value accurately. Hence, the image should be considered LR.

From a more practical point of view, the reference resolution can be defined through an empirical judgement. As the resolution of the input image gradually decreases, the effects of the spatial integration start to become significant. Thus, the reference resolution can be defined as the resolution where these effect start to degrade the fitting
results when using the conventional fitting algorithms. We use this approach to define a reference resolution in our experiments.

4.4 Fitting Algorithm for Low-Resolution

This section presents our proposed approach to fitting a high-resolution 3DMM to a low-resolution 2D image. Similar to the conventional MFF framework, we use multiple cost terms associated with multiple features of the image or the model, and divide our algorithm into multiple stages. At each stage, a combined cost function is formed by a weighted sum of a given set of the cost terms. This cost term is then optimised with respect to the model and/or rendering parameters.

We use the landmarks, image edges, and pixel colour values as features in the MFF framework. A cost function is associated with each of these features. Furthermore, we use two cost terms to account for the shape and texture priors, and a cost term for light regularisation. The landmark cost term as well as the light regularisation term and the prior costs in our method are similar to the conventional MFF algorithm for HR inputs, as was described in Section 2.4, since they do not depend on the resolution of the input. However, for the edge cost and pixel colour features, we use novel cost functions specifically tailored to the case of LR inputs.

In the following, we refer to our LR-specific Fitting algorithm as LRF while the conventional Multi-Feature Fitting algorithm used for HR inputs is referred to as the MFF algorithm.

4.4.1 Pixel Colour Cost

Perhaps the most important and informative criterion for the fitting is the pixel colours. As was argued in Section 4.2 the conventional imaging model is not suitable for describing an LR image. Thus, the conventional pixel colour cost, which is based on this imaging model, becomes sub-optimal in low resolutions. By replacing the conventional imaging model with our low-resolution imaging model (eq. 4.21), we propose a fitting criterion which is suitable for LR images.

The pixel colour cost function aims to maximize the likelihood of the input image given the model and rendering parameters: \( p(I^{inp} | \mu, \rho) \), or equivalently minimising the (negative) log-likelihood \( -\ln p(I^{inp} | \mu, \rho) \). Assuming that the image pixels are affected by independent, identically distributed Gaussian noise, the pixel colour cost function can be expressed as:

\[
-\ln p(I^{inp} | \theta) \propto \sum_{\textbf{m}} \| I^{input}(\textbf{m}) - I^{model}(\textbf{m}) \|^2
\]  

(4.23)

where \( \textbf{m} = [m, n]^T \) is a vector of 2D LR image pixel indices and \( I^{model} \) is given by our LR image formation model as defined in Equation 4.21. Thus, the LR-specific pixel colour cost can be expressed as:
In the above equation, the sum is taken over the pixels, \( \mathbf{m} \), which fall within the face region of the image; where the face region is defined as the region of the image covered by the model, after projecting the model to the image plane\(^3\).

Note that the sum in the cost function of Equation 4.24 is taken over the pixels of the LR input image, as opposed to the conventional cost function used in the SNO and MFF algorithms, Equation 2.25, where the sum is taken over points sampled from the model’s high-resolution texture. In other words, while the conventional approach interpolates the texture of the LR input image to the high-resolution texture of the model, our formulation avoids this interpolation through the LR imaging process. This is in line with the conclusions of Dedeoglu et al. [34] regarding their suggested natural choice for the direction of the warp since our LR image formation model is effectively a blurring and down-sampling process which scales the high-resolution texture of the model to the low-resolution image plane.

Since the pixel colour cost is summed over the pixels of the face on the image plane, it is possible for this cost value to decrease if the model shrinks such that it covers fewer pixels. At the extreme, the value of the cost function can be minimised if the model shrinks such that the projected model only covers a single pixel. Such a situation can normally be prevented by normalising the cost by the number of pixels in the face area (i.e. taking the mean squared error instead of the sum). However, in our framework, this is not necessary since the final cost to be optimised includes contributions from other cost terms as well as the pixel colour term. The contribution of other cost terms, such as the edge cost described in the next section, effectively prevents the model from shrinking artificially.

In the pixel colour cost function described by Equation 4.24, the value of \( A \) represents the total photo-sensitive area of the virtual pixel. This value should normally be the same for all pixels. However, in practice we take this value to be the sum over the overlap areas of all polygons projected to a pixel. Thus the value is different for each pixel:

\[
A(\mathbf{m}) = \sum_{k \in K(\mathbf{m})} W(k, \mathbf{m})
\]  

(4.25)

The set of polygons which overlap with pixel \( \mathbf{m} \), denoted by \( K(\mathbf{m}) \) in the above equations, is calculated at beginning of each stage which uses the pixel colour cost function and updated every few iterations.

\(^3\)In Stage 6 (See Section 4.4.4) of the fitting algorithm, each segment of the face is fitted separately. In this stage, the sum is taken over the area corresponding to the given segment.
4.4.2 Edge Cost

The edge cost function aims to optimise the shape and projection parameters such that the model edges match the edges of the input image. It was discussed in Section 2.4.2, that in the conventional MFF approach a binary edge map is formed by applying the a deterministic edge detector (Canny) to the input image (Equation 2.34). The Chamfer Distance Transform (CDT) is then calculated in order to construct a cost surface which is defined in the same plane and on the same grid as the input image. Subsequently, the edge cost is calculated by projecting each model edge point to the edge cost surface and taking the CDT value at the resulting point:

\[ E_{e}^{\text{MFF}} = \sum_{i} \text{CDT}(e_{i}^{m}(\alpha, \tau))^2 \]  

(4.26)

where \( e_{i}^{m} \) denotes the coordinates of the \( i^{th} \) edge point of the model, after projection to the image plane.

The image edges should ideally be detected with sufficient localisation in order to correspond to fine edges on the model. However, a low-resolution edge pixel would correspond to a much coarser location for the edge. Detecting the edges with the accuracy provided by the LR input image may not provide enough fine detail for the fitting algorithm; thus, it is needed to detect the image edges with sub-pixel accuracy. Moreover, it is impossible for any general edge detection method, even in high resolution, to perfectly find all the edges in a natural image without tuning its parameters to the specific image at hand. For instance, in the case of the Canny edge detector, it is not feasible to find a set of hysteresis thresholds which would be optimal for all input images.

To overcome these problems, we use an approach similar to that of Amberg et al. [5] and extend it to multiple resolutions. We apply edge detection not only at the input resolution, but also at a number of other resolutions, including the reference resolution. Assuming the reference resolution provides sufficiently localised edges, we interpolate the input LR image to multiple resolutions to build a multi-resolution image pyramid where the lowest level (highest resolution) of the pyramid corresponds to the reference resolution. Furthermore, at each resolution we apply multiple edge detectors with different parameters to maximise the possibility of capturing an edge irrespective of the specific input image. The final edge cost surface is generated using information gathered by the set of all the edge detectors across all resolutions and all parameters sets. The stronger edges at each resolution will be detected by most of the parameters sets while some weak edges would still be detected by a few of the detectors. Furthermore, the stronger edges which are consistent across multiple resolutions will be detected at more resolution levels while other edges would be still detected at fewer levels.

Through this procedure, we aim to find edges which are consistent across different resolutions but still be able to include the contribution of weaker edges in the final cost surface. Also, by interpolating the input LR image to higher resolutions and detecting the edges in interpolated images, we aim to sufficiently localise the edges.
Chapter 4. Resolution-Aware Fitting of 3DMM on LR Images

For each set of parameters at each resolution of the pyramid, a separate binary edge map is obtained. Thus, a total of $n_l \times n_p$ edge maps are obtained where $n_l$ is the number of levels in the pyramid and $n_p$ is the number of parameter sets at each level. A separate distance transform (CDT) is calculated for each edge map. Each CDT is separately normalised and upper bound by a threshold in order to avoid bias caused by undetected or spurious edges.

Finally, we integrate information over all configurations of resolution and parameters sets into a smooth edge cost surface, which has the desirable property of having stronger minima at edges over the full range of parameters and resolutions while still discriminating between strong and weak edges.

We use a Gaussian pyramid with 4 levels, $i = 0 \ldots 3$, where $i = 0$ corresponds to the reference resolution. At each resolution we apply $n_p = 11$ Canny edge detectors with different sets of hysteresis thresholds. The complete process to obtain the edge cost surface is as follows:

**Pyramid Formation:** Considering the relative resolution of the input image with respect to the reference frame, the input is placed at level $i = l_{in}$ of the Gaussian pyramid which is $2^{l_{in}}$ times smaller than the reference resolution in each direction. The higher resolutions of the pyramid ($i = l_{in} - 1, \ldots, 0$) are constructed by interpolating the input image. Also, the lower resolutions of the pyramid ($i = l_{in} + 1, \ldots, 3$), if any, are constructed by smoothing and down-sampling the input image.

**Multi-Resolution and Multi-Parameter Edge Analysis:** At each resolution level, $i$, and for each set, $j$, of parameters, the following steps are applied to obtain a normalised CDT:

- Apply the Canny edge detector with the $j^{th}$ set of parameters to the $i^{th}$ level of the pyramid.
- Find the distance transform for this configuration: $CDT_{i,j}$.
- Normalise the distance transform in order to reduce the effect of distant edges while magnifying the effect of closer ones: $\tilde{CDT}_{i,j} = \frac{CDT_{i,j}}{CDT_{i,j} + k_i}$, where the normalising constant $k_i$ is set to approximately $\frac{1}{20}$ of the size of the face at level $i$ of the pyramid.
- Clamp the normalised distance transform to maximum value of 0.5 to obtain the cost surface corresponding to this configuration: $D_{i,j} = \min\left\{\tilde{CDT}_{i,j}, 0.5\right\}$. Note that the value 0.5 in the normalised distance transform corresponds to a distance of $k_i$ pixels from the edge. Therefore, this clamping limits the effective region around each edge to $k_i$ pixels.
4.4. Fitting Algorithm for Low-Resolution Cost Surface Fusion: The final cost surface is obtained by combining the cost surfaces of the different configurations into a single smooth cost surface, $D$. In order to sustain sufficient localisation of the edges, the final combined cost surface should be defined in the image plane, and on the grid of, the reference resolution. Hence, each cost surface $D_{i,j}$ is separately enlarged to the size of the reference resolution by bilinear interpolation before being combined with other cost surfaces. The final combined cost surface can be expressed as:

$$D = \frac{2k_0}{n_l n_p} \sum_{i=0}^{n_l-1} \sum_{j=0}^{n_p-1} \text{Bilin}\{D_{i,j}, 2^i\}$$  \hspace{1cm} (4.27)

where $k_0$ is equal to $\frac{1}{20}$ of the size of the face at the reference resolution, $\text{Bilin}\{D_{i,j}, 2^i\}$ denotes enlarging the 2D surface, $D_{i,j}$, by a factor $2^i$ using bilinear interpolation, and $D_{i,j}$ is the normalised and clamped Chamfer Distance Transform corresponding to the $j^{th}$ set of parameters at the $i^{th}$ resolution level:

$$D_{i,j} = \min \left\{ \frac{\text{CDT}_{i,j}}{\text{CDT}_{i,j} + k_i}, 0.5 \right\}$$  \hspace{1cm} (4.28)

Figure 4.2 compares the edge cost surface (ECS) obtained by our multi-resolution and multi-parameters method (Multi-ECS) with an edge cost surface obtained by the conventional approach (Single ECS).

![Figure 4.2](image)

(a) Single ECS  \hspace{1cm} (b) Multi-ECS

Figure 4.2: Comparison of the Edge Cost Surface obtained with (a) a single edge detector vs. (b) our proposed multi-resolution, multi-parameter approach.

Having found the edge cost surface, the value of the edge cost function for a given model edge point is then obtained by projecting the model edge point to the image plane of this cost surface and taking the cost value at the obtained point. The smooth edge cost surface obtained with this scheme is defined in the image plane corresponding to level zero of the pyramid (reference resolution). In order to project a model edge point to this image plane, we use the vector-valued perspective shape projection function, $p(x, y, z; \tilde{\tau})$, as defined in Section 2.3.1. Here, $(x, y, z)$ are the 3D object-centred coordinates of the model edge point at hand and the vector $\tilde{\tau}$ can be obtained by replacing the focal length, $f$, in the vector of projection parameters, $\tau$, by $2^{\ln f} f$ in order to
account for the resolution difference between the LR input image and the reference resolution. Thus, the value of the edge cost function is given by:

\[ E_{e}^{LRF} = \sum_{i} D(e_{i}^{m}(\alpha, \tilde{\tau}))^2 \]  

(4.29)

where \( D \) is the multi-resolution and multi-parameter edge cost surface given by Equation 4.28 and \( e_{i}^{m} \) is the coordinates of \( i^{th} \) model edge point after projection to the image plane of the reference resolution.

Similar to the conventional MFF approach, we use the orientation of the edge to improve the edge fitting accuracy. At each configuration (resolution and parameter set), the detected edges are divided into four bins according to their orientation and a separate distance transform is calculated for each bin. The distance transforms of each bin are then separately normalised and combined into a cost surface for the corresponding edge orientation. During fitting, the edge cost value for each model edge point is obtained by projecting it to the cost surface corresponding to its orientation.

### 4.4.3 Robust Estimation

The individual cost terms discussed so far have a quadratic form. The problem with a quadratic cost function is that it is not robust to outliers. An outlier, in this context, is an observation value which is numerically distant from the value predicted by the model. Outliers are usually due to features that cannot be explained by the model. For instance, wearing sunglasses would cause an occlusion on the face which is not modelled by the 3DMM. Such an occlusion would result in an outlier with a large residual (i.e. difference between observed and predicted values) for the pixel colours. The problem with the quadratic cost term is that the influence of a particular measurement is proportional to its residual. Hence, the solution can be significantly influenced by outliers with large residuals.

In order to increase the robustness of the algorithm to outliers, we use the Huber cost function \cite{51} instead of the pure quadratic form presented in the previous sections. This cost function has a quadratic form for small residuals and a linear form for larger residuals, thus decreasing the impact of large residuals caused by outliers. The Huber cost function is defined as:

\[ E = \begin{cases} r^2 & |r| \leq \sigma \\ 2\sigma|r| - \sigma^2 & |r| > \sigma \end{cases} \]  

(4.30)

where \( r \) is the residual and \( \sigma \) is the threshold below which the quadratic cost term is used.
4.4.4 Low-Resolution Model Fitting

As was mentioned earlier, our algorithm is divided into multiple stages. At each stage a subset of the sought parameters is fitted through minimisation of a combined cost function using Levenberg-Marquardt optimisation. Table 4.1 presents the features and cost terms used at each stage as well as the parameters fitted. Stage 1 of the algorithm is initialised by average (i.e. 0) shape and texture parameters. The subsequent stages up to stage 5 are then each initialised by the result of the preceding stage. That is, the parameter values estimated in Stage 1 are used to initialise Stage 2 and so on. In Stage 6, however, each segment is fitted separately. Before fitting each segment, the model parameters are re-initialised to some known values which can be either the optimal parameter values obtained at Stage 5, or the mean parameters. Initialising each segment by the results of Stage 5 yields more coherent results for the separate segments as well as more visually pleasing results. On the other hand, re-initialising the parameters to zero prior to fitting each segment increases the independence of the parameter estimates of each segment from other segments and from the global model. We chose the latter approach since the independence of parameters corresponding to different segments is beneficial when using these parameter estimates for recognition.

Table 4.1: List of features and parameters fitted at each stage of the LRF algorithm. On the last line 'segm.' stands for segmented, highlighting the fact that at this stage each segment of the model is fitted separately. In Stage 2, the first three alphas are initialised using only the anchor points. In Stages 3 and 4, only the first 20% of the $\alpha$ and $\beta$ parameters are optimised. Considering that we have 55 shape and 123 texture parameters, this means that in Stage 3, only the first 11 shape ($\alpha$) parameters are optimised and in Stage 4, only the first 25 texture ($\beta$) parameters.

<table>
<thead>
<tr>
<th>Stage Nb.</th>
<th>Features</th>
<th>Parameters</th>
<th>Nb. of Par.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 X</td>
<td>X</td>
<td>X</td>
<td>6</td>
</tr>
<tr>
<td>2 X</td>
<td>X</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3 X X X</td>
<td>X</td>
<td>X X 11</td>
<td>17</td>
</tr>
<tr>
<td>4 X X X X</td>
<td>X X 25</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>5 X X X X</td>
<td>X X X X X</td>
<td>186</td>
<td></td>
</tr>
<tr>
<td>6 X X X</td>
<td>segm.</td>
<td>segm. 712</td>
<td></td>
</tr>
</tbody>
</table>

In addition to the sub-optimality of the cost function for low-resolution images, which was discussed in previous sections, the optimisation methods used for optimising the cost function in the fitting algorithms of [86] and [20] are not suitable for the low-resolution case. Assuming that the contributions of all pixels of the image to the overall cost are redundant, these methods used a stochastic optimisation scheme which only evaluates the cost over a small number of vertices in each iteration. The reason for using the stochastic optimisation was to gain efficiency and avoid local minima at the cost of limited convergence properties (such as convergence radius).

For a low-resolution input however, the initial assumption of redundant contributions from image pixels is no longer valid. In fact, due to the lack of information in a low-resolution image it is crucial to ensure that all available information is used. Hence, we do not use a stochastic optimisation algorithm. Instead, we deal with the problem of local minima by using a multi-feature fitting strategy similar to that of [90]. Due to using multiple features the overall cost function in this framework is smoother and a stochastic optimisation is not necessary to avoid local minima ([88]). This means that
all polygons of the model are used in evaluating the pixel colour cost function and the sum in Equation 4.24 is taken over all LR pixels. This makes the computational cost of our algorithm relatively high compared to conventional HR fitting algorithms.

4.5 Experimental Evaluation

This section presents an experimental evaluation of our proposed LR fitting approach. We compare the performance of our approach to the conventional MFF\(^4\) approach over a range of different resolutions using a subset of the PIE database.

The original images are at high resolution. Through low-pass filtering and down-sampling, we generate a set of low-resolution images at different resolutions from each original image. In the experiments presented here, we have used between 10 to 14 landmarks to initialise the fitting. These landmarks are manually annotated on the original HR images. Landmarks for the LR images are then obtained by scaling the coordinates of the original HR landmarks. Note that automatically detecting accurate landmarks on low-resolution faces is a challenging task which is outside the scope of this thesis. However, it was observed in our experiments that the accuracy of the landmarks did not have a significant impact on the final result. In other words, both fitting algorithms (MFF and LRF) proved to be fairly robust to small errors in landmark positions. Roughly speaking, the fitting algorithms can tolerate errors of about 7% of the inter-eye distance without a significant impact on the final result\(^5\). Recall from Table 4.1 that the landmark cost is only used in the first two stages of the algorithm for initialisation and is not used in any of the later stages. Hence, small misalignment errors due to landmark errors can still be corrected by later stages of the algorithm.

Figures 4.3 to 4.5 illustrate, for three different subjects, some of the LR images produced with different down-sampling factors (DSF), as well as the model fitting results using the conventional MFF and our proposed approach (LRF) approach. While the conventional fitting fails to recover detailed texture for down-sampling factors larger than 4, our approach manages to recover a reasonable amount of the detail even in much lower resolutions, notably outperforming the conventional method at lower resolutions. An interesting observation to be made is in Figure 4.5 at DSF = 12 where the conventional approach has even failed to correctly align the model with the face.

As was mentioned in the previous section, the computational cost of our algorithm is high. The most time-consuming part of the optimisation in any given iteration is calculation of the Jacobian matrix for the combined cost function. This matrix includes the first derivatives of the combined cost function with respect to each of the parameters fitted at the given stage. Thus, the computational cost of each stage depends on the cost terms used in the given stage as well as the number of parameters optimised.

\(^4\)The specific implementation of the MFF framework used in our experiments was developed at the Centre for Vision, Speech, and Signal Processing according to the original publications of Romdhani et al. [88, 90]

\(^5\)This arguments should only be regarded as an observation as opposed to a verified conclusion. More experiments would be required to quantitatively determine the fitting robustness with respect to the accuracy of the landmarks.
Among the different cost terms used in our framework, the pixel colour cost term is by far the most computationally complex cost term. Calculation of the pixel colour cost and its derivatives plays an important role in increasing the computation time of each iteration in the stages which make use of this cost term. It should be noted that the total time spent in each stage depends not only on the computational complexity of each iteration, but also on the number of iterations required for convergence which in turn depends on the initial conditions.

Table 4.2 shows an example of the times spent in each of the fitting stages for the image in Figure 4.3 (for DSF=8). The first 3 stages of the fitting require less computation time since these stages do not use the pixel colour cost which is the most computationally expensive cost term in our framework. Stage 4 uses this cost term and therefore is more computationally involved and more time-consuming compared to the previous 3 stages. However, since only a subset of the parameters are fitted in Stage 4, the computational cost of this stage is less than Stage 5 which uses the pixel colour cost term and optimises all the parameters. In stage 6, all the shape and texture parameters are optimised for each segment separately. This stage is also among the more time-consuming stages of the fitting algorithm since it uses the pixel colour cost term. Note that among the various segments fitted in Stage6, the “rest” segment requires more time for convergence. This is due to the fact that this segment is considerably larger than the other 3 segments. Hence, more polygons and more pixels need to be evaluated. Also, note that the “rest” region requires more time than Stage 5 despite the fact that Stage 5 optimises more parameters and fits the whole face. This is due to the manner in which these stages are initialised. Stage 5 is initialised by the results of Stage 4 which means that this stage is typically initialised closer to the optimum. However, each segment in Stage 6 is initialised by the average model which means that the initial point is potentially further from the optimum and the algorithm would require more iterations for convergence.

<table>
<thead>
<tr>
<th>Stage</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>eyes</td>
<td>nose</td>
<td>mouth</td>
<td>rest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time (s)</td>
<td>0.64</td>
<td>3.16</td>
<td>3.17</td>
<td>16.03</td>
<td>58.31</td>
<td>37.59</td>
</tr>
</tbody>
</table>

Once the model fitting has been performed, one can use the recovered parameters to render the face in any desired resolution, pose, or illumination. Figure 4.6 shows an example of the model fitted to an LR image with $DSF = 8$ and rendered at the resolution equivalent to $DSF = 2$, in other words, 4 times enlarged. For comparison, we have also included the results of the same rendering when the model was fitted using the conventional MFF algorithm, as well as bilinear interpolation of the LR image. Note that the conventional algorithm has completely failed in fitting the model when the input resolution is very low (the input resolution in this figure is equivalent to the third column of Figure 4.3). However, the proposed LRF approach has successfully recovered much of the high-resolution texture and clearly provides much more detail than what was present in the LR image, represented by the bilinear interpolated image. The extra HR information added to the image is provided by the prior knowledge of the HR face space, captured by the HR samples used for building the 3DMM.
Figure 4.3: Comparison of model fitting on LR images. Each column shows, from left to right, images with DSF = 4, 6, 8, and 12 respectively. Top row: original image, middle row: Model fitted using the conventional MFF algorithm. Bottom row: Model fitted using our LRF approach.
Figure 4.4: Comparison of model fitting on LR images. Each column shows, from left to right, images with DSF=4,6,8, and 12 respectively. Top row: original image, middle row: Model fitted using the conventional MFF algorithm. Bottom row: Model fitted using our LRF approach.
Figure 4.5: Comparison of model fitting on LR images. Each column shows, from left to right, images with DSF=4, 6, 8, and 12 respectively. Top row: original image, middle row: Model fitted using the conventional MFF algorithm. Bottom row: Model fitted using our LRF approach. Note that for DSF = 12, the conventional approach has even failed to align the model with the face correctly.
4.5. Experimental Evaluation

Figure 4.6: Enlarging the face after fitting the model.
Besides the visual inspection of the fitting results, we evaluate the performance of our proposed approach quantitatively by using the fitted model to render an image of the face and measuring the similarity of this rendered image with the original input over the face region. We select a subset of the CMU-PIE dataset consisting of two poses (poses 05 and 27) and 3 different illuminations (illuminations 01, 02, and 13) for this evaluation. This subset is formed of 408 images in total which provides sufficient variation over different subjects, poses, and illumination conditions. For each image from the subset, we generate a set of down-sampled input images with a range of down-sampling factors ($DSF = 1, 2, 4, 6, 8, 12, 16$). We then fit the model using our proposed approach and the conventional MFF approach to all the images and compare the rendered image with the input in each case.

The image-domain similarity between the input face and the rendered face is expressed in terms of two different similarity measures, namely the Peak Signal-to-Noise Ratio (PSNR) and the Mutual Information (MI) between the two images.

PSNR is a frequently-used image quality measure which measures the ratio between the maximum power of a signal and the power of the noise affecting it. In our case, the signal is the original input face and we assume that the rendered face is a noisy estimate of the input. PSNR can be calculated as:

$$PSNR = 10 \log_{10} \left( \frac{MAX^2}{MSE} \right)$$

where $MAX = 255$ is the maximum possible range of the signal, and $MSE$ is the Mean Squared Error between the two images, over the face region. Denoting the input image by $I^{inp}$, the rendered image by $I^{ren}$, and the area of the image covered by the face by $\mathcal{F}$, the Mean Squared Error can be calculated as:

$$MSE = \frac{1}{3N_{\mathcal{F}}} \sum_{i \in \mathcal{F}} [I^{inp}(i) - I^{ren}(i)]^2$$

where $N_{\mathcal{F}}$ is the number of pixels in the face region. Note that the number of pixels is multiplied by 3 which is the number of colour channels (RGB) in the image. Figure 4.7 compares the PSNR values, averaged over all 408 samples, for both fitting methods over the range of down-sampling factors considered in this experiment.

As can be seen in Figure 4.7, both approaches yield similar PSNR values for the $DSF = 1$ (original resolution) and $DSF = 2$. However, the PSNR values corresponding to the conventional MFF approach decrease rapidly with decreasing resolution of the input image while the values corresponding to the proposed low-resolution fitting approach show much higher consistency across the range of down-sampling factors considered. These results confirm that the model fitted using our proposed LRF approach resembles the input LR images more accurately compared to the model fitted using the conventional MFF approach.

As mentioned earlier, we also use Mutual Information (MI) as another similarity measure. Mutual Information between two random variables can be intuitively interpreted...
as the amount of information that knowing either variable provides about the other. More specifically, MI between two variables $X$ and $Y$ is a measure of the inherent dependence expressed in the joint distribution of the two variables relative to the same joint distribution under the assumption of independence:

$$MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 \left\{ \frac{p(x, y)}{p(x)p(y)} \right\}$$ (4.33)

In the case of our experiment, the random variables $X$ and $Y$ are vectors formed by pixel values over the face area in the input and the rendered images, respectively. Figure 4.8 shows the average MI values obtained in our experiment over a range of different resolutions. Again, it can be observed that when using the conventional MFF, the similarity of the two images decreases rapidly with decreasing resolution of the input image. However, the values obtained when using the proposed LRF algorithm show much higher similarity between the two images in lower resolutions which suggests that the model fitted using the LRF approach bears much more resemblance to the input image compared to the model fitted using the conventional approach.

Finally, we compare the performance of the proposed approach with the conventional MFF algorithm in the model space, ie. in terms of the recovered parameters. We measure accuracy of the fitting as the similarity between the model parameters recovered by fitting to an LR image and those that would have been recovered if the input image was HR. Recall that the aim of our proposed algorithm is face recognition in a scenario where the gallery images are HR and the probe images are LR. Hence, the aim is to recover parameters from the LR image which are similar to their HR counterparts. We use the same subset of the PIE dataset as the one used in the previous experiment. The model parameters obtained using the conventional fitting on the original HR images ($DSF = 1$) are taken as ground truth. Note that this is a plausible choice of ground truth since a) The true 3D parameters for these real images are not known; b) The similarity of the HR parameters to the true 3D parameters is outside the scope of this
Chapter 4. Resolution-Aware Fitting of 3DMM on LR Images

Figure 4.8: Average MI measured over 408 samples for a range of down-sampling factors: $DSF = 1, 2, 4, 6, 8, 12, 16$. Error margins show one standard deviation.

paper and has been addressed in other works (e.g. [86] [88]); and c) In a realistic scenario, these HR parameters are the ones that would be used for most applications (e.g. face recognition) where the input is a 2D image.

Performance evaluation is carried out by measuring the similarity of LR parameters, obtained by model fitting at different low resolutions, to the ground-truth HR parameters. Ideally, the parameters recovered from a low-resolution image should be identical to those that would have been obtained if the input image had a high resolution. We measure the similarity in parameter space in terms of Normalised Correlation (NC) which measures the cosine of the angle between two parameter vectors. The NC similarity score between the HR shape vector, $\alpha_{HR}$, and the LR shape vector, $\alpha_{LR}$, is given as:

$$NC(\alpha_{HR}, \alpha_{LR}) = \frac{\alpha_{HR}^T \alpha_{LR}}{\sqrt{\alpha_{HR}^T \alpha_{HR}} \sqrt{\alpha_{LR}^T \alpha_{LR}}}$$  (4.34)

The NC similarity score between the texture vectors, $\beta_{HR}$ and $\beta_{LR}$, can be expressed similarly. Figure 4.9 compares the similarity scores for shape and texture vectors obtained using our algorithm with those obtained using conventional MFF. Values shown are the average NC scores over 408 samples together with error bars showing one standard deviation.

It is clear from Figure 4.9 that our algorithm performs significantly better than the conventional MFF in low resolutions. Note that in higher resolutions ($DSF = 2$), the conventional MFF outperforms our method. This is expected since our algorithm is specifically designed for low resolutions where numerous polygons are projected to the same image pixel. In HR images where this assumption does not hold our algorithm performs worse than the conventional MFF due to its higher ambiguity. However, since during the early stages of the fitting it can easily be confirmed whether the input image is HR or LR (for instance, by considering the estimated focal length and distance to
4.6 Discussion and Conclusions

In the previous chapters of this thesis, we reviewed the 3DMM as a tool for unconstrained face recognition under varying poses and illuminations. By explicitly modelling these extrinsic phenomena, the 3DMM is able to account for the variations they cause. However, when it comes to low-resolution inputs, the conventional approaches to fitting a 3DMM fail to perform acceptably.

Through a critical analysis of the fitting criteria commonly used in the conventional algorithms, we argued that the reason for this decrease in performance is that the imaging model used by such criteria is not suitable for modelling the process of forming an LR image and is only valid under the implicit assumption that the input image has a high resolution.

Inspired by the super-resolution methods reviewed in Chapter 3, we proposed a suitable LR imaging model which, unlike the conventional imaging model, takes into account the virtual camera’s point spread function and the spatial integration over the LR pixels. We used this imaging model to formulate a pixel colour cost function which forms the core of our LR fitting algorithm.

In addition, we proposed an enhanced method for extracting a smooth cost surface for the edge cost when the input image is LR. Detecting edges in LR images is a challenging task. Furthermore, when edges are detected in an LR image, they don’t provide sufficient localisation for fitting the fine model edges. We address these problems by proposing to detect edges not only at the input resolution but also at multiple other resolutions including the reference resolution, thus providing a more robust and accurate edge cost surface with sufficient localisation. At each resolution multiple edge detectors are used with different sets of parameters to maximise the chances of detecting all possible edges. By fusing the information obtained at different resolutions and

Figure 4.9: Similarity between HR and LR model parameters in the parameter space in terms of NC scores. Right: shape similarity, Left: texture similarity. Error margins show one standard deviation.

camera), it is straightforward to propose a hybrid algorithm which uses the conventional MFF for HR inputs and switches to LRF if the input is LR.

4.6 Discussion and Conclusions

In the previous chapters of this thesis, we reviewed the 3DMM as a tool for unconstrained face recognition under varying poses and illuminations. By explicitly modelling these extrinsic phenomena, the 3DMM is able to account for the variations they cause. However, when it comes to low-resolution inputs, the conventional approaches to fitting a 3DMM fail to perform acceptably.

Through a critical analysis of the fitting criteria commonly used in the conventional algorithms, we argued that the reason for this decrease in performance is that the imaging model used by such criteria is not suitable for modelling the process of forming an LR image and is only valid under the implicit assumption that the input image has a high resolution.

Inspired by the super-resolution methods reviewed in Chapter 3, we proposed a suitable LR imaging model which, unlike the conventional imaging model, takes into account the virtual camera’s point spread function and the spatial integration over the LR pixels. We used this imaging model to formulate a pixel colour cost function which forms the core of our LR fitting algorithm.

In addition, we proposed an enhanced method for extracting a smooth cost surface for the edge cost when the input image is LR. Detecting edges in LR images is a challenging task. Furthermore, when edges are detected in an LR image, they don’t provide sufficient localisation for fitting the fine model edges. We address these problems by proposing to detect edges not only at the input resolution but also at multiple other resolutions including the reference resolution, thus providing a more robust and accurate edge cost surface with sufficient localisation. At each resolution multiple edge detectors are used with different sets of parameters to maximise the chances of detecting all possible edges. By fusing the information obtained at different resolutions and
by different detectors, we obtain a smooth edge cost surface which provides consider-ably more information about the edges of the input image while providing sufficient localisation for model fitting.

Our fitting approach can be compared to some of the approaches to LR face analysis presented in Chapter 3. Among the categories reviewed in Chapter 3, our approach falls within the “model-based” category. Perhaps the most closely related LR face analysis method in the literature to our approach is the resolution-aware AAM fitting approach of Dedeoglu et al. [33]. Similar to our approach they used an LR imaging model which describes the process of forming an LR image given the model parameters. They formulate the AAM fitting criterion such that the HR model is warped to the LR input image via the LR imaging model, thus avoiding interpolation of the input image. Their imaging model and the resulting criterion are comparable to our LR imaging and pixel cost term. Furthermore, the model space is used to include prior knowledge in the fitting procedure. Obviously, the main difference is that they used a 2D AAM model while our model is a 3D model. As a result, we are able to model different poses and also model scene illumination more explicitly. Moreover, they only used the pixel colours while we also take advantage of other features, mainly the image edges.

Our approach can also be compared to the example-based approaches to super-resolution which use a set of exemplar face images to learn prior information about a human face and use it in a MAP estimation framework.

Such methods use two main sources of information:

- **Reconstruction constraint**: The reconstruction constraint leads to an LR imaging model widely used in the super-resolution literature. This model is generally used to formulate the likelihood term in the MAP estimation framework.

- **Prior information**: Using the sample set, prior information about a face is learnt and used to formulate the prior term in a MAP estimation framework.

In comparison, our framework learns 3D information and uses it to model 2D inputs via the fitting procedure. The fitting procedure is in fact an analysis by synthesis loop where the synthesis step is responsible for projecting the 3D information of the model to the 2D image space. This is done through an LR imaging model which satisfies the reconstruction constraint. The synthesised image obtained through our 3D-to-2D LR imaging model is then used to formulate a cost term corresponding to the likelihood term in a MAP framework. Furthermore, the model provides prior information about a human face which is used in the fitting process to formulate the prior term. However, unlike most of the example-based super-resolution methods, our approach is able to generalise to previously unseen poses and illuminations by explicitly accounting for variations caused by such extrinsic factors.

Since our framework directly infers HR information (HR model) from LR inputs, it can also be considered a direct HR-LR matching approach. In other words, using our model-based approach for face recognition, both the HR gallery and the LR probe images will eventually be described in the same face space described by the model.
Experimental evaluation showed that our proposed fitting approach can successfully fit the 3DMM to low resolution images when the conventional MFF algorithm fails. We provided fitting results for visual inspection and also evaluated the similarity of the fitted model to the input image in terms of two popular measure, namely, PSNR and MI. The results of these comparisons confirm that the proposed low-resolution fitting approach significantly outperforms the state-of-the-art conventional fitting approach which fails at low resolutions.

Furthermore, we compared the performance of the proposed approach with the conventional approach in the parameter space of the model. Model parameters obtained by fitting the model on a low-resolution image should be similar to those that would have been obtained if the input image was high-resolution. Ideally, these should be the same. Comparison of the fitted model parameters using Normalised Correlation confirmed that the parameters obtained with the proposed approach are much more similar to their HR counterparts over a considerably large range of resolutions.

Through the discussions and experiments presented in this chapter, we conclude that the proposed method is suitable for model fitting on LR face images. In the next Chapter we will show how the proposed method can be used in a recognition scenario for low-resolution face recognition under varying poses and illuminations.
Chapter 5

3DMM for Unconstrained Face Recognition in Low Resolution

5.1 Introduction

We presented an approach for fitting a 3DMM to LR images in the previous chapter. In this Chapter we will show how the resulting model parameters can be used to perform pose- and illumination-invariant face recognition in low-resolution. As discussed earlier in Chapter 2, two different approaches can be taken in order to perform face recognition using a 3DMM. The first approach is to use the model parameters directly for recognition and the second approach is to use the model in order to render a normalised 2D image which will then be used in a conventional 2D face recognition system. In this chapter, we take the former approach, i.e. we will use the model parameters directly for recognition. We will use the latter approach in the experiments presented in the next chapter.

In the following sections, we first describe the details of a face recognition method that uses model parameters for recognition in Section 5.2 and present a systematic evaluation of face identification performance with our framework using synthetically degraded images in Section 5.3. In Section 5.4, a novel method is presented based on Linear Discriminant Analysis (LDA) as an alternative method of using the model parameters for face recognition. The performance of this method in a face verification scenario is evaluated experimentally and it is shown that the combination of our proposed LRF model fitting algorithm and the proposed LDA-based method can yield considerably high performance in LR face verification. Section 5.5 summarises and concludes this chapter.

5.2 Using Model Parameters for Face Recognition

Given a 2D image, the model fitting process yields two types of parameters describing, respectively, the shape and texture of a face. Together, these parameters provide sufficient information for identifying the face.
Chapter 5. 3DMM for Face Recognition in Low Resolution

Recall from Chapter 2 that the fitting procedure optimises the global model in Stages 1 to 5 and in the final stage four regions are fitted separately. Hence, 5 separate sets of shape and texture parameters are obtained through this procedure. One set corresponding to the global model and four sets corresponding to each of the four segments. In addition, we obtain another set of parameters by reconstructing a 3D face from each of the regional parameter sets, blending the 4 faces into a single 3D face, and projecting the result into the model space. Thus, 6 sets of shape and texture parameters are obtained. All these parameters are then used to build an identity vector which is used for recognition.

The simplest approach to building a descriptive feature vector from the shape and texture parameter vectors is to normalise each parameter by the respective standard deviation, and concatenate all 6 sets of shape and texture parameters into an identity vector which holds all necessary information for recognition:

\[ \zeta = \left[ \hat{\alpha}_g^T, \hat{\alpha}_{s_1}^T, \hat{\alpha}_{s_2}^T, \hat{\alpha}_{s_3}^T, \hat{\alpha}_{s_4}^T, \hat{\alpha}_b^T, \hat{\beta}_g^T, \hat{\beta}_{s_1}^T, \hat{\beta}_{s_2}^T, \hat{\beta}_{s_3}^T, \hat{\beta}_{s_4}^T, \hat{\beta}_b^T \right]^T \] (5.1)

where the superscript \( g \) denotes the global model, superscripts \( s_1 \) to \( s_4 \) denote the four segments of the model, and superscript \( b \) denotes the parameters obtained by blending the four segments into a single 3D face and projecting to the model’s PCA space. In Equation 5.1, \( \hat{\alpha}_g \) is the normalised global shape vector:

\[ \hat{\alpha}_g = \left[ \frac{\alpha_{g_1}}{\sigma_{S,1}}, \frac{\alpha_{g_2}}{\sigma_{S,2}}, \ldots, \frac{\alpha_{g_{n_\alpha}}}{\sigma_{S,n_\alpha}} \right]^T \] (5.2)

where \( n_\alpha = 55 \) is the number of model shape parameters\(^1\) and \( \sigma_{S,j} \) is the standard deviation of the \( j^{th} \) shape parameter. Similarly, \( \hat{\beta}_g \) is the normalised global texture vector:

\[ \hat{\beta}_g = \left[ \frac{\beta_{g_1}}{\sigma_{T,1}}, \frac{\beta_{g_2}}{\sigma_{T,2}}, \ldots, \frac{\beta_{g_{n_\beta}}}{\sigma_{T,n_\beta}} \right]^T \] (5.3)

with \( n_\beta = 123 \) denoting the number of model texture parameters and \( \sigma_{T,j} \) denoting the standard deviation of the \( j^{th} \) texture parameters. The other normalised parameter vectors in Equation 5.1 are obtained similarly to Equations 5.2 and 5.3 for the four segments and the blended model.

For the purpose of face recognition, the 3DMM is fitted to each gallery and probe image and the corresponding identity vector is formed using Equation 5.1. The similarity between the probe identity vector and each gallery identity vector is then measured. We use Normalised Correlation (NC) between the identity vectors as the similarity measure. The NC similarity measure is insensitive to the norm of the identity vectors.

\(^1\)Note that \( n_\alpha \) and \( n_\beta \) were denoted by \( D_s \) and \( D_T \), respectively, in Chapter 2.
This is favourable for the recognition task as the effect of increasing the norm of the identity vector is largely seen as producing a caricature of the face and not changing the perceived identity [88].

5.3 Face Identification

Face identification is the problem of identifying an individual from a dataset of known identities, referred to as the gallery set. The probe image is compared to all images in the gallery set. A similarity measure (or equivalently, a distance measure) is assigned to each pair of gallery and probe images, describing the similarity of the probe image to the given gallery image. The identity of the gallery image which is most similar to the probe image (highest similarity; or equivalently, lowest distance) is chosen as the identification outcome.

Let us denote the identity vector of the probe image by $\zeta^P$, and that of the $c^{th}$ gallery image by $\zeta^{Gc}$. As mentioned previously, the similarity measure used in our experiments is the normalised correlation between the two identity vectors. The identification outcome is then defined as the identity of the gallery image which yields the highest NC score. That is, identity $\hat{c}$ is assigned to the probe image at hand, where:

$$\hat{c} = \arg\max_c \{NC(\zeta^P, \zeta^{Gc})\}$$ (5.4)

We use images from the CMU-PIE dataset in our face identification experiments to evaluate the performance of our proposed approach over a range of different resolutions, poses and illuminations. This dataset includes 68 subjects and contains, for each subject, multiple images at different poses and illuminations. The dataset contains images with the room light on or off. Since the images taken with the room lights on appear more natural and more representative of images in the real world, we only use this subset of the dataset in our experiments. This subset contains images with three different poses per subject, as illustrated by Figure 5.1. These images are taken simultaneously using different cameras. Each image is identified by a number representing the camera which was used for taking the corresponding image. Accordingly, the frontal, side, and profile poses are identified by camera numbers 27, 5, and 22, respectively.

![Figure 5.1: Different poses present in the PIE dataset.](image)

In addition to pose variations, the CMU-PIE dataset contains multiple images with variations in illumination conditions for each subject. 24 images with different illumination conditions are available per subject per pose. The first two images are taken
without flashes, leaving only the ambient light to illuminate the face. These are identified as illumination 00 and 01. Then 21 images are taken, each with one flash light firing to illuminate the face from a particular direction in addition to the ambient light. These are identified by illuminations 2 to 22. Sample images of different illumination conditions for the “side” pose are shown in Figure 5.2. The flash number appears below each image. Finally, the last image, identified by illumination 23, is taken without flash similar to the first two images.

Figure 5.2: Illuminated images in the PIE dataset. The flash numbers are given underneath each image. Note that the images illuminated only by ambient light (Illumination 00, 01, and 23) are not shown. (Figure taken from [88])

The images in the PIE dataset are high-resolution. The average inter-eye distance of the frontal samples is 86 pixels. In the following experiments, we generate multiple different resolutions by down-sampling the original images with different down-sampling factors (DSF).

5.3.1 Finding the Reference Resolution

As mentioned in Section 4.3.4, the reference resolution is defined as the conceptual threshold between high and low resolutions. We find this resolution empirically for the PIE dataset. To this end, we perform an identification experiment in which the resolution of the gallery and probe sets is gradually reduced and the identification performance is measured at each resolution. It is expected that, from some point on, reducing the resolution would start affecting the recognition performance. We take this point as the reference resolution.

For the purpose of this experiment, we take frontal images (pose 27) with ambient plus frontal illumination (illumination 08) as the gallery set while taking frontal images with only ambient illumination (illumination 00) as the probe set. Table 5.1 shows the rank-1 recognition rates at different down-sampling factors (DSF) used to down-sample the original images. Note that in this experiment both gallery and probe are down-sampled
so they are always at the same resolution. Therefore, the performance variation across the different resolutions is only due to the absolute resolution of the images and not the resolution difference between gallery and probe.

Table 5.1: Rank-1 Identification Rate at different resolutions of the gallery and probe images.

<table>
<thead>
<tr>
<th>DSF</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification Rate</td>
<td>95.6</td>
<td>95.6</td>
<td>76.5</td>
<td>64.7</td>
<td>61.8</td>
</tr>
</tbody>
</table>

The results of Table 5.1 show that the performance at $DSF = 2$ is equal to the performance at $DSF = 1$ (original images); in other words, the recognition performance is not affected by down-sampling the gallery and probe images by a factor of 2. However, the performance drops rather significantly when the images are further down-sampled ($DSF = 4$). Since this variation in performance is merely caused by a change in resolution, it can be concluded that at this resolution ($DSF = 4$) the effects of low-resolution have become significant enough to affect the model fitting performance which in turn degrades the recognition performance.

From the above discussion we conclude that the reference resolution for this database can be set at $DSF = 2$. In other words, images down-sampled by a factor of 2 can still be regarded as HR while any further down-sampling should be regarded as LR. In the following experiments involving the PIE dataset we use $DSF = 2$ as the reference resolution. Accordingly, the HR gallery set consists of images down-sampled by a factor of 2.

### 5.3.2 The gallery

Choosing the right gallery set is an important task in face recognition. The gallery images should be chosen such that they represent each subject accurately. Gallery samples should contain sufficient discriminatory information for each enrolled individual. In a typical face recognition scenario, the gallery images are taken under controlled imaging conditions. Thus, it can be assumed that the gallery images are high-resolution images with a suitable illumination and pose. It is commonly considered that a frontal image is most suitable for gallery since it provides minimal occlusion in the areas of the face that are important for recognition. Furthermore, to prevent extrinsic variations caused by illumination conditions from affecting the gallery set, the gallery images are usually assumed to be taken under uniform illumination.

The images in the PIE dataset contain 3 poses; namely frontal, side, and profile. As discussed above, it would seem a natural choice to take the frontal image with a uniform illumination as the gallery. For instance, the frontal images with only ambient illumination (illumination 00) or with frontal illumination (e.g. Illumination 08, 11, 20) would be a natural choice for the gallery. However, considering that our approach relies on 3D information extracted from the 2D images, the frontal images may not be the best option. More specifically, while the frontal pose provides minimal occlusion in the most important areas of the face, it may not optimally provide 3D depth information. On the other hand the side pose images of the dataset are taken at an angle of about
Chapter 5. 3DMM for Face Recognition in Low Resolution

30 degrees. This pose provides more 3D details of the shape. Therefore, model fitting to these images results in a more accurate estimate of the shape which in turn can improve the texture estimate. Thus, one can expect this pose to be an equally suitable choice, or even a more suitable one, for the gallery set.

We verify the above claim by performing a simple experiment to compare rank-1 recognition results for each gallery option. More specifically, we take the frontal pose as the gallery set and find the rank-1 recognition rate for both frontal and side probe images. We then take the side pose as gallery and repeat this experiment. For fair comparison, for each gallery pose we have chosen an illumination condition which illuminates the visible area of the face rather uniformly. We chose illumination 08 for the frontal gallery and illumination 13 for the side gallery. In all cases, the probe images were chosen with ambient-only illumination.

Table 5.2 compares the rank-1 identification rates for different configurations of gallery and probe poses. It can be seen that the performance in the cases where side pose is used for gallery is comparable or even superior to the performance in cases where the frontal pose is used. This experiment suggests that either one of these two poses could be used for the gallery set. In the following, we provide three experiments using the PIE dataset which use the frontal and/or the side pose for gallery.

Table 5.2: Rank-1 Identification Rate using different configurations of gallery and probe poses.

<table>
<thead>
<tr>
<th>Gallery</th>
<th>Probe</th>
<th>Frontal</th>
<th>Side</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frontal</td>
<td></td>
<td>95.59</td>
<td>82.35</td>
<td>88.97</td>
</tr>
<tr>
<td>Side</td>
<td></td>
<td>82.35</td>
<td>97.06</td>
<td>89.70</td>
</tr>
</tbody>
</table>

The following sub-sections will present experiments to evaluate the performance of our model fitting approach for face recognition under varying illumination and/or poses. The first two experiments, presented in Sections 5.3.3 and 5.3.4, evaluate the performance under varying illuminations and poses, respectively, over a range of different resolutions. Having demonstrated the trend of the performance over a range of different resolutions using these experiments, our third experiment, presented in Section 5.3.5, considers the case where both pose and illumination conditions vary simultaneously.

5.3.3 Illumination-Invariant LR Face Identification

We present an experiment to evaluate the performance of our approach under different illumination conditions over a range of different resolutions. We use a subset of the PIE dataset with frontal pose and a representative set of illumination conditions. More specifically, the gallery set is a frontal image with only ambient illumination (illumination 00) while the probe set consists of frontal images with four different illumination conditions as shown in Figure 5.3.

We evaluate the identification performance for a range of different resolutions by down-sampling the probe set and performing the identification experiment at each resolution. Figure 5.4 and Table 5.3 show the rank-1 recognition rates at different down-sampling
5.3. Face Identification

Factors for each of the illumination conditions and compare the performance of our proposed approach (LRF) with that of the conventional model fitting approach (MFF).

As illustrated by the plots of Figure 5.4, the performance of the MFF approach is significantly degraded as the resolution of the input image is reduced. The average rank-1 identification rate in high resolution ($DSF = 2$) for the MFF algorithm was 97.79 (not shown in the plots). This rate drops down to 92.28 when the resolution of the probe set is halved ($DSF = 4$). Any further reduction in the probe set resolution ($DSF = 6, 8, \ldots$) results in significant degradation in the performance of the system. On the other hand, the proposed LRF approach is significantly more robust to changes in the resolution of the probe image and shows a considerably higher performance at low resolutions when compared to the conventional MFF method. Note that although the performance of our proposed method also degrades with decreasing resolution, this performance degradation happens at a much slower pace compared to the conventional approach.

Furthermore, it can be concluded from Figure 5.4 and Table 5.3 that the improved performance in LR is obtained in all illumination cases. Higher identification rates are obtained irrespective of whether the illumination conditions of the probe and gallery images are similar (Figure 5.4(a)) or not. Thus, the proposed LRF provides an illumination-invariant method for face recognition in low-resolution.

Figure 5.3: The probe set includes a representative set of different illumination conditions: a) Ambient only; b) Strong side illumination; c) Frontal illumination; d) Mild side illumination. The gallery set consists of frontal images with ambient illumination similar to “a”.
Figure 5.4: Rank-1 identification results under changing illumination over a range of resolutions. The probe set consists of: a) Ambient illumination; b) Strong side illumination; c) Frontal illumination; d) Mild side illumination. The gallery set consists of frontal images with ambient illumination similar to “a”.

Table 5.3: Rank-1 identification results under changing illumination over a range of resolutions. Values in bold show the performance of the proposed LRF method while values in parentheses show the performance of the conventional MFF approach for comparison.
5.3.4 Pose-Invariant LR Face Identification

The next experiment evaluates the performance of our proposed approach for different poses over a range of resolutions. For the gallery set we use images with side pose (Pose 05) and Illumination 13 which provides uniform illumination for the side pose. The probe set consists of images with the same illumination condition at three different poses, namely, frontal, side, and profile. Figure 5.5 shows samples of the probe set.

![Probe Set Samples](image)

(a) Pose 05 (side)  
(b) Pose 22 (profile)  
(c) Pose 27 (frontal)

Figure 5.5: The probe set includes images with ambient illumination and different poses: a) Side; b) Profile; c) Frontal. The gallery set consists of frontal images with ambient illumination, similar to “c”.

Again, we perform the experiment at multiple different resolutions by down-sampling the probe set, and we compare the performance of our proposed approach with the conventional fitting approach at each resolution. Figure 5.6 and Table 5.4 show the rank-1 identification rates for each pose over different resolutions.

Figure 5.6 and Table 5.4 show that our proposed method outperforms the MFF algorithm over the range of low resolutions considered for all considered probe poses. For the frontal and side poses, remarkably high identification rates are obtained with the LRF method even at very low resolutions (e.g. $DSF = 12$). In comparison, both methods show lower performance for the profile pose (Figure 5.6(b)). However, note that even for this extreme pose, the performance of our proposed method is considerably better than the conventional MFF algorithm for resolutions down to $DSF = 8$.

These experiments confirm that the proposed fitting algorithm yields improved performance irrespective of the pose, and thus, offers a pose-independent framework for face identification in low resolutions.
Figure 5.6: Rank-1 identification results under changing pose over a range of resolutions. The probe set consists of: a) Side pose; b) Profile pose; c) Frontal pose. The gallery set consists of frontal images with ambient illumination, similar to “c”.

Table 5.4: Rank-1 identification results under changing poses over a range of resolutions. Values in bold show the performance of the proposed LRF method while values in parentheses show the performance of the conventional MFF approach for comparison.

<table>
<thead>
<tr>
<th>Probe Pose</th>
<th>DSF</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>12</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>05 (side)</td>
<td>100</td>
<td>100</td>
<td>97.06</td>
<td>98.53</td>
<td>85.29</td>
<td>85.29</td>
</tr>
<tr>
<td>05 (side)</td>
<td>100</td>
<td>100</td>
<td>97.06</td>
<td>98.53</td>
<td>85.29</td>
<td>85.29</td>
</tr>
<tr>
<td>22 (profile)</td>
<td>82.35</td>
<td>63.23</td>
<td>63.23</td>
<td>29.41</td>
<td>36.76</td>
<td>36.76</td>
</tr>
<tr>
<td>22 (profile)</td>
<td>76.47</td>
<td>57.35</td>
<td>36.76</td>
<td>17.64</td>
<td>13.23</td>
<td></td>
</tr>
<tr>
<td>27 (frontal)</td>
<td>100</td>
<td>98.53</td>
<td>100</td>
<td>95.59</td>
<td>66.18</td>
<td></td>
</tr>
<tr>
<td>27 (frontal)</td>
<td>95.59</td>
<td>80.88</td>
<td>60.29</td>
<td>22.06</td>
<td>7.35</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>94.12</td>
<td>87.25</td>
<td>86.76</td>
<td>74.51</td>
<td>62.7451</td>
<td>62.7451</td>
</tr>
<tr>
<td>Average</td>
<td>90.69</td>
<td>78.92</td>
<td>58.33</td>
<td>24.51</td>
<td>12.2549</td>
<td>12.2549</td>
</tr>
</tbody>
</table>
5.3.5 Pose- and Illumination-Invariant Face Identification

The next experiment evaluates the face identification performance under conditions where the pose and illumination of the probe set both differ simultaneously from those of the gallery set. Two separate experiments are performed, one with frontal images as gallery and the other with the side pose images used for gallery. In both cases the gallery contains images with ambient-only illumination (Illumination 00).

The probe sets consist of all other combinations of poses and illuminations present in the dataset, that is, three poses and 23 illumination conditions. Low resolution probe images are generated by down-sampling the original HR images by a factor of 8.

Table 5.5 summarises the average rank-1 identification rates over all considered illuminations, for each gallery and probe pose.

Table 5.5: Rank-1 identification performance for different gallery/probe pose combinations. The reported results are averaged over all possible illumination conditions.

<table>
<thead>
<tr>
<th>Probe pose</th>
<th>Gallery Pose</th>
<th>Frontal</th>
<th>Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frontal</td>
<td>89.13</td>
<td>84.97</td>
<td></td>
</tr>
<tr>
<td>Side</td>
<td>76.21</td>
<td>88.04</td>
<td></td>
</tr>
<tr>
<td>Profile</td>
<td>72.83</td>
<td>69.25</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>79.39</td>
<td>80.75</td>
<td></td>
</tr>
</tbody>
</table>

The results presented in Table 5.5 demonstrate that the proposed LRF method is able to extract sufficient information from the LR images for recognition in this large range of varying imaging conditions. The lowest identification rates are obtained for the profile probe set using either frontal or side gallery. This is expected due to the extreme pose in which only half of the face is visible, limiting the available information for model fitting.

Both experiments with the frontal and side gallery images demonstrate relatively similar performance when averaged over all illumination conditions and all three poses, with the average rank-1 identification rate at 79.39% for the frontal gallery set and 80.75% for the side gallery.

The same-pose performance of the frontal gallery set is slightly better than the side pose. That is, when the the gallery and probe sets are both frontal, the rank-1 identification rate is at 89.13% whereas this rate is slightly lower at 88.04% when side-pose images are used for both gallery and probe sets.

On the other hand, when the pose of the gallery and probe sets don’t match, the side-pose gallery set demonstrates relatively higher performance. More specifically, the average performance of the frontal gallery set over the side and profile probe sets is 74.52%, whereas the average performance of the side gallery over frontal and profile probe sets is 77.11%.

The performance of the method in each of the considered illumination conditions is summarised in Table 5.6. In this table the rank-1 identification rate is averaged over 3 probe poses at each of the illumination conditions.
The performance is generally lower for the cases where the probe image is lit by a strong flash light from the side. More specifically, the identification rates obtained for Illumination 2, 5, 10, 18, are consistently lower than the average performance in both cases where frontal or side galleries are used. However, there are exceptions such as Illumination 03, in which the performance is higher than average in both cases and Illumination 04, where the performance is higher than average when using the side gallery set.

Table 5.6: Rank-1 identification rates of the proposed LRF method for LR probe images (DSF=8) at different illumination conditions, averaged over three probe poses.

<table>
<thead>
<tr>
<th>Probe Illumination</th>
<th>Frontal</th>
<th>Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>84.80</td>
<td>81.37</td>
</tr>
<tr>
<td>02</td>
<td>75.49</td>
<td>78.43</td>
</tr>
<tr>
<td>03</td>
<td>79.90</td>
<td>82.84</td>
</tr>
<tr>
<td>04</td>
<td>75.98</td>
<td>81.86</td>
</tr>
<tr>
<td>05</td>
<td>74.51</td>
<td>76.96</td>
</tr>
<tr>
<td>06</td>
<td>76.96</td>
<td>81.37</td>
</tr>
<tr>
<td>07</td>
<td>75.49</td>
<td>80.39</td>
</tr>
<tr>
<td>08</td>
<td>76.96</td>
<td>80.39</td>
</tr>
<tr>
<td>09</td>
<td>76.47</td>
<td>78.43</td>
</tr>
<tr>
<td>10</td>
<td>75.98</td>
<td>76.96</td>
</tr>
<tr>
<td>11</td>
<td>81.86</td>
<td>82.35</td>
</tr>
<tr>
<td>12</td>
<td>84.31</td>
<td>84.31</td>
</tr>
<tr>
<td>13</td>
<td>80.39</td>
<td>82.84</td>
</tr>
<tr>
<td>14</td>
<td>84.31</td>
<td>82.35</td>
</tr>
<tr>
<td>15</td>
<td>80.39</td>
<td>81.86</td>
</tr>
<tr>
<td>16</td>
<td>81.37</td>
<td>81.86</td>
</tr>
<tr>
<td>17</td>
<td>80.39</td>
<td>76.47</td>
</tr>
<tr>
<td>18</td>
<td>78.43</td>
<td>79.90</td>
</tr>
<tr>
<td>19</td>
<td>76.47</td>
<td>80.39</td>
</tr>
<tr>
<td>20</td>
<td>81.37</td>
<td>78.43</td>
</tr>
<tr>
<td>21</td>
<td>77.94</td>
<td>83.33</td>
</tr>
<tr>
<td>22</td>
<td>82.84</td>
<td>80.88</td>
</tr>
<tr>
<td>23</td>
<td>83.33</td>
<td>83.33</td>
</tr>
<tr>
<td>Average</td>
<td>79.39</td>
<td>80.75</td>
</tr>
</tbody>
</table>

5.4 Face Verification

Person identification, presented in the previous section, involves a one-to-many matching where the probe sample is compared against the whole gallery set. Person verification, on the other hand, involves a one-to-one matching. In face verification, a person submits a facial image and claims an identity. Two kinds of claims can be considered. We refer to these as genuine and imposter claims. A genuine claim is the case where a person enrolled in the system, known as a client, claims his/her own identity. An
imposter claim, on the other hand, is the case where a subject who is not enrolled in
the system, known as an imposter, claims the identity of one of the enrolled clients.
The task of a verification system is to verify whether the person is who he/she claims to
be. In other words, the task is to accept the genuine claims and to reject the imposter
claims. To this end, the system produces an access score for the given claim, which is
essentially a similarity score describing the similarity of the submitted probe sample
with the gallery sample(s) corresponding to the claimed identity. This similarity score
is then compared against the system’s operating threshold. If the similarity score is
higher than this threshold, the system will accept the claim as a genuine one, thus
verifying that the person is who they claim to be. Otherwise, the claim is rejected as
an imposter claim.

We use the XM2VTS dataset [74] to evaluate the performance of our proposed frame-
work in face verification. This dataset includes images of 295 subjects taken in 4
different recording sessions. We use a set of images from the dataset which includes 2
frontal images per session per subject as well as 4 rotation shots, each with a distinct
non-frontal pose. The rotation shots include, for each subject, 4 images in which the
subject is looking to the left, right, up, and down, respectively.

For the task of person verification, a standard protocol [75] has been defined for the
XM2VTS dataset. The so-called Lausanne protocol splits all subjects randomly into a
client and an imposter set. The client set includes 200 subjects while the imposter set
contains 25 evaluation imposters and 70 test imposters. The protocol only covers the
frontal images. All available frontal images are divided into a training, an evaluation,
and a test set. Two configurations exist in the protocol which differ in the particular
selection of each person’s images into the training, evaluation and test sets. Here, we
use the first configuration in which 3 training shots, 3 evaluation shots, and 2 test shots
are available per client. Figure 5.7 illustrates the particular partitioning into these sets.
The training set is primarily intended to be used for building client models, while the
evaluation set is used for setting the operating threshold of the system and the test set
is used for measuring the system’s performance. In addition to the 2 frontal shots used
in the test set according to this protocol, we also use the 4 rotation shots as test data
in order to evaluate the performance of our proposed approach for non-frontal probe
images.

The performance of a verification system is typically measured via two commonly-
used error measure: False Acceptance Rate (FAR) and False Rejection Rate (FRR).
False acceptance is the case where an imposter, claiming the identity of a client, is
accepted. False rejection is the case where a client, claiming the true identity, is rejected.
Accordingly, the aforementioned error rates are defined as follows:

\[ \text{FAR} = \frac{N_{AT}}{N_T} \times 100, \quad \text{FRR} = \frac{N_{RC}}{N_C} \times 100 \]  \hspace{1cm} (5.5)

where \( N_{AT} \) is the number of accepted imposter claims, \( N_T \) is the total number of imposter
claims, \( N_{RC} \) is the number of rejected client (genuine) claims, and \( N_C \) is the total
number of client (genuine) claims.
5.4.1 LDA-Based Face Verification

We described a simple approach to face recognition using the model parameters in Section 5.2 which uses the model parameters to form an \textit{identity vector} and establishes similarity between individuals by comparing their identity vectors. In this section we propose an alternative approach using Linear Discriminant Analysis (LDA) [12].

As mentioned in Section 5.2, the fitting process yields 6 sets of shape and texture parameters for each image. Thus, a total of 12 separate parameter vectors are available, for each face, as a result of model fitting. Assuming sufficient training data is available, we propose to calculate a separate LDA transformation for each of these parameter vectors:

\[
\chi^k = W^k \nu^k
\]  

(5.6)

where \( \nu^k \) for \( k = 1, 2, \ldots, 12 \) is the \( k^{th} \) parameter vector obtained by model fitting, \( W^k \) is the corresponding LDA transform, and \( \chi^k \) is the projection of this vector to the corresponding LDA space. The similarity between a pair of parameter vectors after projection to the LDA space is measured using normalised correlation:

\[
s_{P, G_j^c}^k = NC(\chi_P^k, \chi_{G_j^c}^k)
\]

(5.7)

where the subscript \( P \) refers to the probe sample and the subscript \( G_j^c \) refers to the \( j^{th} \) gallery sample\(^2\) of client \( c \). Accordingly, \( s_{P, G_j^c}^k \) in Equation 5.7 denotes the similarity between the \( k^{th} \) parameter vectors of these two samples. 12 such similarity scores are obtained for a given pair of gallery and probe samples. The final similarity score between the probe sample, \( P \), and the gallery sample, \( G_j^c \), is obtained by a weighted sum of all 12 scores obtained by this approach:

\(^2\)Note that the gallery may contain more than one sample per client.
The weights, $a^k$, of this sum reflect one’s confidence in a given similarity score. The absolute values of the weights can be determined by an empirical judgement. However, certain relationships between the weights can be envisaged based on one’s prior knowledge of the expected descriptive information provided by each set of parameter vectors. In particular, we expect the parameters corresponding to the whole face to be more descriptive compared to those parameters which correspond to a specific segment. Accordingly, we assign lower weights to similarity scores corresponding to segment-specific vectors, $\hat{\alpha}^s_i$ and $\hat{\beta}^s_i$ ($i = 1, 2, 3, 4$), while assigning higher weights to similarity scores corresponding to the global parameter vectors of Stage 5, $\hat{\alpha}^g$ and $\hat{\beta}^g$, as well as the parameter vectors obtained by blending the segments, $\hat{\alpha}^b$ and $\hat{\beta}^b$. Moreover, texture parameters prove to be more descriptive than shape parameters, especially in the low-resolution case where shape estimates can be unreliable. Therefore, we assign higher weights to texture similarity scores and lower weights to shape similarity scores.

Considering the above discussion, there’s a scope for appropriately optimising the weights in Equation 5.8, using the evaluation set. However, in this work, we only coarsely sampled the space of weights in order to demonstrate the idea. In the experiments presented in the next section, we used $a^k = 4$ and $a^k = 1$ for $k$ values corresponding to the global and local shape parameters vectors, respectively, and $a^k = 30$ and $a^k = 7$ for $k$ values corresponding to the global and local texture parameter vectors, respectively.

### 5.4.2 Experimental Setup and Results

Following the Lausanne protocol three frontal samples per client are available as the training data. We use this data for training the 12 LDA transforms. To simulate a realistic LR face verification scenario, we only use high-resolution frontal images for training. To this end, the 3DMM is fitted to each of the HR training images using the conventional MFF approach. After model fitting, each training sample is represented by 12 parameter vectors. The LDA transforms are trained using this data, where the $k^{th}$ LDA transform, $W^k(k = 1, \ldots, 12)$, is trained using the $k^{th}$ parameter vectors of all 600 training images.

The evaluation set consists of $200 \times 3 = 600$ client images and $25 \times 8 = 200$ imposter images. We use the evaluation set in order to produce client and imposter access scores, which are then used to find a threshold that determines if a person is accepted or not. Again, in order to simulate a realistic scenario, we only use HR frontal images in the evaluation set. Each of the 600 client samples in the evaluation set only makes genuine claims by claiming its own identity. Each imposter sample, on the other hand, claims all possible identities in the gallery set. Thus, 600 genuine claims and 40 thousand imposter claims are possible.
When a person claims a given identity, the probe sample submitted by the claimer is compared against all gallery samples corresponding to the claimed identity. In order to perform the comparison between the probe sample and a given gallery sample, the 3DMM is fitted to both images using the conventional MFF algorithm. The 12 parameter vectors obtained for each sample are then projected to their corresponding LDA spaces and a similarity score is calculated for this image pair using the weighted sum rule, as discussed in Section 5.4.1. Considering that there are three samples per identity in the gallery set, each claim results in three comparison scores, each of which is the result of fusing 12 similarity scores using the weighted sum rule (Equation 5.8). A further level of score fusion is used to fuse the three scores obtained for each claim. We use the min rule for this fusion. That is, when a probe sample claims the identity of a given subject, we compare the probe sample at hand with all three gallery samples available for the claimed identity and take the minimum of the similarity scores as the final access score for the claim at hand:

\[ s_{P,Gc} = \arg\min_j \{ s_{P,Gc_j} \} \]  

(5.9)

where the index \( j \in \{1, 2, 3\} \) runs over the training samples for client \( c \) and \( s_{P,Gc} \) denotes the access score for the case where probe sample \( P \) claims the identity of client \( c \).

The access scores produced in the evaluation step are used to set the operating threshold of the system. We set this threshold such that the false acceptance and false rejection rates over the evaluation set are equal. This is referred to as the Equal Error Rate (EER) threshold:

\[ T_{EER} = \arg\min_T \{ T \mid \text{FAR}_{\text{eval}}(T) = \text{FRR}_{\text{eval}}(T) \} \]  

(5.10)

where \( T_{EER} \) is the EER threshold and \( \text{FAR}_{\text{eval}}(T) \) and \( \text{FRR}_{\text{eval}}(T) \) are, respectively, the false acceptance and false rejection rates, obtained using threshold \( T \), over the evaluation set.

Figure 5.8 illustrates the FAR and FRR rates with respect to \( T \). The EER threshold, \( T_{EER} \), is the value of \( T \) at which the two plots cross. In our experiment, the EER threshold is at \( T_{EER} = 0.0135 \) with the evaluation Equal Error Rate at this threshold equal to 5.01%.

The EER threshold given by Equation 5.10 is then used in the test step in order to accept or reject the test claims. Similar to the evaluation step, each client sample only makes genuine claims while each imposter sample claims all possible identities in the gallery. Again, each claim results in 3 similarity scores, \( s_{P,Gc_j} \), corresponding to comparison of the submitted probe sample with each of the 3 gallery samples of the claimed identity. This time, we use majority voting to make the final acceptance or rejection decision for a given claim. That is, a claim is accepted if and only if \( s_{P,Gc_j} > T_{EER} \) for at least 2 out of 3 possible values of \( j \).
5.4. Face Verification

Figure 5.8: Plots showing the False Acceptance Rate and False Rejection Rates with respect to the operating threshold $T$. At a very low threshold all client and imposter samples are accepted. Thus, $FRR = 0$ and $FAR = 100$. At a very high threshold, all samples would be rejected resulting in $FRR = 100$ and $FAR = 0$. The EER point is where the two errors are equal.

Performance of the verification system over the test set is measured by means of the Half-Total Error Rate (HTER) which is the average of FAR and FRR errors over the test set:

$$HTER = \frac{FAR_{test} + FRR_{test}}{2}$$  \hspace{1cm} (5.11)

Table 5.7 summarises the Half-Total Error Rates obtained with our approach over the different test subsets and compares these with the values for the case where the conventional MFF approach was used for fitting the model on test samples. Note that the training and evaluation steps are the same for both cases and the difference between the presented results is only in the fitting method used in the test stage where the probe images are LR.

Table 5.7: Half-Total Error Rates over the test set. Both methods achieve acceptable results when using the LDA-based verification approach, but the proposed LRF method outperforms the conventional MFF method in all poses.

<table>
<thead>
<tr>
<th>Method</th>
<th>Front</th>
<th>Left</th>
<th>Right</th>
<th>Up</th>
<th>Down</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMF+LDA</td>
<td>8.43%</td>
<td>13.53%</td>
<td>11.89%</td>
<td>17.4%</td>
<td>10.2%</td>
<td>12.29%</td>
</tr>
<tr>
<td>LRF+LDA</td>
<td>6.77%</td>
<td>9.29%</td>
<td>8.94%</td>
<td>14.74%</td>
<td>9.26%</td>
<td>9.8%</td>
</tr>
</tbody>
</table>

The proposed LDA-based recognition approach yields reasonable results even in the case where the conventional MFF algorithm was used for model fitting on the LR test samples. However, the best performance was consistently obtained for all probe poses when the proposed LRF fitting algorithm and the LDA-based recognition methods were used. This confirms that our proposed LRF method is more efficient in extracting useful
information from LR inputs than the conventional MFF approach, and when combined with the proposed LDA-based recognition approach, it can yield a considerably high verification performance over different poses.

The results obtained for the “up” pose show lower performance compared to other poses in both methods. This can be associated with the inherent ambiguity in this particular pose. The model seems to be unable to decide whether the image is a person looking upwards or a person with a smaller face looking forward. Note that this is an inherent ambiguity of the model and is independent of the image input resolution. Addressing this ambiguity is outside the scope of this thesis.

5.5 Summary and Conclusion

This chapter presented and evaluated the application of our LR-specific model fitting approach to face recognition in low-resolution images. Two scenarios of person identification and person verification were considered. Each scenario was evaluated using a different dataset and it was shown that our proposed approach can achieve high performance rates in both cases.

In the case of person identification, a complete evaluation over a range of different resolutions was performed in two different scenarios where the pose or illumination of the probe set were different from those of the gallery. This evaluation over a range of different resolutions confirms that our approach shows significantly higher robustness to decreased probe resolution compared to the conventional MFF approach which suffers from a considerable performance degradation in low resolutions.

A more realistic scenario of person identification was also presented in which both pose and illumination of the probe set differed from those of the gallery set. The results of this experiment showed that our LRF approach can be used successfully for pose- and illumination-independent face recognition in low-resolution.

Furthermore, we proposed an LDA-based approach for face verification using the model parameters obtained by model fitting. We used this approach in a face verification experiment using the XM2VTS dataset. It was shown that face verification performance can be enhanced by using the LRF method for model fitting on LR probe face images. This result was consistent over different frontal and non-frontal poses which once again shows the suitability of our framework for pose-independent face recognition in low-resolution.

From the experiments presented in this chapter, it is concluded that the proposed framework can enhance low-resolution face recognition performance in all cases of variations in pose and illumination. However, the enhanced performance is limited in extreme poses, extreme illuminations, and for extremely low resolutions (eg. DSF=16).
Chapter 6

A 3D-Assisted Facial Texture Super-Resolution Framework

6.1 Introduction

In Chapter 3, different categories of approaches to face recognition in low-resolution were reviewed. One of the most common categories of approaches to address this problem is using super-resolution (SR) as a pre-processing step to enhance the resolution of the probe image prior to recognition. Super-resolution methods aim to reconstruct the high-frequency texture detail which is missing in an LR image. The high-frequency detail injected back into the LR image by super-resolution, increases the discriminatory information available in the image. Hence, using SR as a pre-processing step can significantly improve the performance of a conventional face recognition engine. Note that most conventional methods of 2D image-based face recognition rely on the facial texture for extracting discriminatory information and their performance would be significantly compromised if the texture lacks high-frequency discriminatory detail.

We reviewed multiple super-resolution algorithms in Section 3.2 and argued that example-based methods are the most suitable category of methods for the task of face super-resolution. This is due to the fact that, unlike reconstruction-based methods, the example-based methods don’t only rely on the reconstruction constraint as the sole source of information. These methods also take advantage of a set of exemplar images to learn prior information which they use as an additional source of information. The use of this additional source of information enables example-based super-resolution methods to perform single-frame super-resolution. More importantly, it enables these methods to overcome the theoretical limits from which reconstruction-based methods suffer [7]. However, the down side of using a set of exemplar images to learn prior information is that the learnt information is limited by the sample set. Consequently, example-based methods are, in general, limited to imaging conditions (poses, illuminations, etc.) represented in the sample set and their ability to generalise to previously unseen imaging conditions is limited.

We discussed in the previous chapters how, given a 2D face image as input, a 3D morphable model can be used to separate the effects of extrinsic factors such as pose
and illumination from intrinsic factors (shape and texture). In this chapter we present a framework to use this information in order to overcome the limitations of example-based super-resolution. More specifically, we use the information extracted from a 2D image by the 3DMM in order to generate a normalised texture map in which the effects of facial shape, pose and scene illumination are normalised. This enables us to generalise the information represented by a set of exemplar images and use it to super-resolve facial texture in previously unseen poses and illumination conditions.

Section 6.2 presents the proposed framework and formulation as well as some results and suggestions for further improvements. This framework is then used in Section 6.3 for face recognition in low-resolution. Finally, we present a discussion of the proposed framework in Section 6.4 and conclude this Chapter in Section 6.5.

6.2 Proposed Framework

We discussed in the previous chapters that, given a 2D image, a 3D Morphable Face Model can be used to estimate facial shape and texture as well as rendering parameters which reflect the imaging conditions. It was also discussed, in Section 2.5, how the original facial texture from the input image can be extracted and mapped to a normalised texture map in order to have the option of using the original facial texture, where available. Such a texture map provides a pose- and shape-normalised representation of the facial texture. It is on this texture image that we propose to perform example-based super-resolution. That is, we first build a texture sample set by fitting a 3DMM on a set of exemplar HR facial images and extracting texture from them. Then, given an input LR image, we super-resolve its texture by fitting the model on the input LR image, extracting the texture, and super-resolving the extracted texture. The super-resolved texture can then be used together with the estimated shape parameters and arbitrary rendering parameters to render a new HR image of the face in the same or a different pose. Although the estimated shape parameters represent a low-resolution shape, the final rendered image will have a high-resolution appearance since it is rendered by mapping HR texture. Figure 6.1 illustrates this process.

6.2.1 Formulation

We formulate the problem as finding the HR facial texture, given an LR image. Let \( T \) denote HR texture represented in the shape- and pose-normalised texture coordinate frame described in Section 2.5. The sought HR texture map, \( T^* \), is the texture map that maximises the following marginalised probability distribution:

\[
T^* = \arg\max_T \sum_{\mu, \rho} p(T, \mu, \rho|L)
\]

\[
= \arg\max_T \sum_{\mu, \rho} \left[ p(T|\mu, \rho, L)p(\mu, \rho|L) \right] \tag{6.1}
\]
6.2. Proposed Framework

Figure 6.1: The proposed framework for 3D-Assisted Facial Texture Super-Resolution.

where \( L \) is the LR input image, \( \mu \) is a set of 3DMM parameters (\( \mu = \{\alpha, \beta\} \)), and \( \rho \) is a set of rendering parameters, reflecting the imaging conditions (\( \rho = \{\tau, \gamma\} \)).

The above equation can be simplified by assuming that \( p(\mu, \rho|L) \) peaks at the optimal values of the model and rendering parameters and it has a dense distribution around these values. In other words, we assume that the distribution of \( p(\mu, \rho|L) \) can be estimated by \( \delta(\mu - \mu^*)\delta(\rho - \rho^*) \) where \( \delta(.) \) is the Dirac delta function, and \( \mu^* \) and \( \rho^* \) are the optimum model and rendering parameters, respectively:

\[
\{\mu^*, \rho^*\} = \arg\max_{\mu, \rho} \{p(\mu, \rho|L)\} \tag{6.2}
\]

Given the above assumption, Equation 6.1 simplifies to:

\[
T^* = \arg\max_T \{p(T|\mu^*, \rho^*, L)\} \tag{6.3}
\]

where \( \mu^* \) and \( \rho^* \) are given by 6.2. Recall from the previous chapters that the optimal model parameters are obtained by fitting the 3DMM to the input image, \( L \). In fact, Equation 6.2 is the original formulation of the model fitting problem. Hence, the first step for optimising Equation 6.1 is fitting the model to the input image, \( L \), in order to obtain the optimal 3DMM parameters, \( \mu^* \) and \( \rho^* \).

Having found the optimal 3DMM parameters, the original facial texture available in \( L \) can be extracted and mapped to the normalised texture coordinate frame. Let us denote this LR texture map by \( t \). Note that the texture mapping process is a deterministic process. Hence, \( t \) is a deterministic function of the model parameters, \( \mu^* \), rendering parameters, \( \rho^* \), and the LR input, \( L \):

\[
t = \text{TEXTURE\_EXTRACT} (\mu^*, \rho^*, L) \tag{6.4}
\]
Chapter 6. A 3D-Assisted Facial Texture Super-Resolution Framework

Considering that the inputs of the \textsc{Texture\_Extract}(.) function fully describe its output and that the output is unique for a given set of inputs, Equation 6.3 can be further simplified by replacing the distribution of \( p(T|\mu^*, \rho^*, L) \) with \( p(T|t) \):

\[
\mathcal{T}^* = \arg\max_{\mathcal{T}} \{ p(T|\mu^*, \rho^*, L) \} = \arg\max_{\mathcal{T}} \{ p(T|t) \} = \arg\max_{\mathcal{T}} \{ p(t|\mathcal{T})p(\mathcal{T}) \} \quad (6.5)
\]

Equation 6.5 describes finding a high-resolution texture map given a low-resolution one, i.e. texture super-resolution.

Following the above discussion, the process of finding \( \mathcal{T}^* \) from Equation 6.1 can be addressed in three steps:

1. **Model Fitting**: The optimal values of the 3DMM parameters together with the optimal values of rendering parameters are obtained by fitting the model to the input image (Equation 6.2).

2. **Texture Extraction**: Facial texture from the LR input face is extracted and mapped to the texture domain (Equation 6.4).

3. **Texture Super-resolution**: Super-resolution in the texture domain is formulated as a MAP estimation problem (Equation 6.5).

We use the Low-Resolution Fitting (LRF) method proposed in Chapter 4 for fitting the 3DMM to the LR image in the first step and the isomap-based method described in Section 2.5 for texture extraction in the second step.

The texture map built in the second step represents the facial texture in a shape- and pose-normalised coordinate frame. However, the effects of illumination would still be present in this texture map. Considering that the model fitting process also provides information about scene illumination, we use this information to normalise the illumination with the aim of recovering face albedo from the extracted texture. To this end, we assume that the extracted texture, \( t \), is the result of illuminating the facial albedo by the Phong reflectance model, followed by linear transformation of this illuminated texture by a colour transformation similar to the one in Equations 2.20 and 2.21. Given this illuminated and colour-transformed texture and having estimated the necessary parameters through the model fitting process, we attempt to de-illuminate the facial texture by reversing the illumination and colour transform processes.

After de-illumination, a representation of the LR facial albedo is available which is normalised for facial shape, subject pose, and scene illumination. Without loss of generality, and for notational convenience, we use the same notation, \( t \), to refer to the de-illuminated texture. From this point on, the term \textit{texture map} and the notation \( t \) are used to refer to the de-illuminated facial albedo map, unless explicitly mentioned otherwise.

The third step of our framework is texture super-resolution. Similar to some of the image super-resolution algorithms reviewed in Section 3.2, the texture super-resolution problem in our 3D-assisted framework is formulated as a MAP estimation problem:
6.2. Proposed Framework

\[ T^* = \arg\max_{T} \{ p(T|t) \} \]
\[ = \arg\max_{T} \{ p(t|T)p(T) \} \]
\[ = \arg\min_{T} \{ -\ln[p(t|T)] - \ln[p(T)] \} \]

(6.6)

Assuming that the LR texture map, \( t \), is a low-pass filtered and down-sampled version of the sought HR texture map, \( T \), we use the reconstruction framework, proposed by Baker and Kanade [10], to recover the HR texture map. Recall from Section 3.2.4 that this is an example-based MAP estimation algorithm for object-specific super-resolution. The low-pass filtering and down-sampling process which yields the LR texture map given the sought HR texture map, is described by a generative model:

\[ t(m,n) = \frac{1}{w^2} \sum_{(p,q) \in \text{bin}(m,n)} T(p,q) + \eta(m,n) \]

(6.7)

where \( w \) is the down-sampling factor, \( \eta \) is a texel noise term, \((m,n)\) refers to an LR texel, \((p,q)\) refers to an HR texel, and \( \text{bin}(m,n) = \{(p,q) \mid \lfloor \frac{p}{w} \rfloor = m, \lfloor \frac{q}{w} \rfloor = n\} \) is the set of all HR texels that are mapped to the LR texel \((m,n)\) after down-sampling. The generative model of Equation 6.7 describes each LR texel as the average of all HR texels that are mapped to it. The difference between this generative model and the image observation model in Section 3.2.1 is that this model does not include registration since the texture maps are, by definition, assumed to be aligned. Given the above generative model, and assuming an \( i.i.d \) Gaussian distribution for the texel noise, (negative log of) the likelihood term in Equation 6.6 is given as:

\[ -\ln[p(t|T)] = \frac{1}{\sigma^2} \sum_{(m,n)} \left[ t(m,n) - \frac{1}{w^2} \sum_{(p,q) \in \text{bin}(m,n)} T(p,q) \right]^2 \]

(6.8)

where \( \sigma^2 \) is the covariance of the texel noise distribution.

As mentioned in Section 3.2.4, the prior term in the reconstruction framework is a recognition-based prior which is defined by finding, for each input pixel, the most similar pixel from a sample set. We follow a similar approach in order to define the prior term, \( p(T) \), in Equation 6.6. First, we build a texture sample set by fitting the 3DMM to a set of exemplar HR face images, extracting texture from them, and de-illuminating the extracted texture. Let us denote the \( i^{th} \) HR texture map extracted from the \( i^{th} \) exemplar image by \( T^i_s \). The extracted texture maps are used to build an \( N \)-level

---

1Recall from Chapter 3 that this method performs super-resolution using the reconstruction constraint and a recognition-based prior. The name reconstruction signifies the two main sources of information used by this framework.

2We refer to pixels of the texture map as texels in order to distinguish them from image domain pixels.

3The MFF algorithm is used for fitting the 3DMM on HR sample images.
Gaussian pyramid [23] of textures, \( G_0(T^s_i), G_1(T^s_i), \ldots, G_N(T^s_i) \), where the \( l^{th} \) level of the pyramid is defined as:

\[
G_l(T^s_i) = \begin{cases} 
T^s_i 
& \text{if } l = 0 \\
2 \downarrow [g \otimes G_{l-1}(T^s_i)] 
& \text{otherwise}
\end{cases}
\]  

(6.9)

where \( 2 \downarrow \) denotes the down-sampling of an image by a factor of 2 in each dimension, \( g \) is a low-pass filter\(^4\), and \( \otimes \) is the convolution operation.

Using this Gaussian pyramid, the Laplacian pyramid [24] of textures, \( L_0(T^s_i), L_1(T^s_i), \ldots, L_N(T^s_i) \), is formed, in which level \( l \) of the pyramid is defined as:

\[
L_l(T^s_i) = \begin{cases} 
G_l(T^s_i) - \text{EXPAND}(G_{l+1}(T^s_i)) 
& \text{if } l \neq N \\
\text{EXPAND}(G_0(T^s_i)) 
& \text{if } l = N
\end{cases}
\]  

(6.10)

where \( \text{EXPAND}(I) \) enlarges image \( I \) by a factor of 2, using pixel replication:

\[
\text{EXPAND}(I)(m, n) = I([\frac{m}{2}], [\frac{n}{2}])
\]  

(6.11)

Furthermore, we form pyramids of the horizontal, \( H_0(T^s_i), H_1(T^s_i), \ldots, H_N(T^s_i) \), and vertical, \( V_0(T^s_i), V_1(T^s_i), \ldots, V_N(T^s_i) \), first derivatives as well as the horizontal, \( H_0^2(T^s_i), H_1^2(T^s_i), \ldots, H_N^2(T^s_i) \), and vertical, \( V_0^2(T^s_i), V_1^2(T^s_i), \ldots, V_N^2(T^s_i) \) second derivatives of the Gaussian pyramid.

Collectively, the above feature pyramids are used to build the \textit{parent structure} vector [32] for each of the sample texture maps. The \( l^{th} \) level parent structure vector, \( \text{PS}_l \), at pixel \((m, n)\) of an image \( I \) is defined as the \( 5 \times (N + 1 - l) \) dimensional vector:

\[
\text{PS}_l(I)(m, n) = [L_l(I)([\frac{m}{2}], [\frac{n}{2}]), L_{l+1}(I)([\frac{m}{2}, \frac{n}{2}], [\frac{m}{2}, \frac{n}{2}]), \ldots, L_N(I)([\frac{m}{2N}], [\frac{n}{2N}]), H_l(I)([\frac{m}{2}], [\frac{n}{2}]), H_{l+1}(I)([\frac{m}{2}, \frac{n}{2}], [\frac{m}{2}, \frac{n}{2}]), \ldots, H_N(I)([\frac{m}{2N}], [\frac{n}{2N}]), V_l(I)([\frac{m}{2}], [\frac{n}{2}]), V_{l+1}(I)([\frac{m}{2}, \frac{n}{2}], [\frac{m}{2}, \frac{n}{2}]), \ldots, V_N(I)([\frac{m}{2N}], [\frac{n}{2N}]), H_l^2(I)([\frac{m}{2}], [\frac{n}{2}]), H_{l+1}^2(I)([\frac{m}{2}, \frac{n}{2}], [\frac{m}{2}, \frac{n}{2}]), \ldots, H_N^2(I)([\frac{m}{2N}], [\frac{n}{2N}]), V_l^2(I)([\frac{m}{2}], [\frac{n}{2}]), V_{l+1}^2(I)([\frac{m}{2}, \frac{n}{2}], [\frac{m}{2}, \frac{n}{2}]), \ldots, V_N^2(I)([\frac{m}{2N}], [\frac{n}{2N}])]
\]  

(6.12)

The parent structure, as defined above, provides a local, multi-scale description of the facial texture at each texel of the sample set.

Let \( L \) be the input LR face image which is \( w = 2^l \) times smaller than the HR images used for the sample set, and \( t \) be the LR texture map extracted from \( L \). The amount of information available in \( t \) is equivalent to the \( l^{th} \) level of the sample set Gaussian pyramid. Hence, given each texel of the input LR texture map, we form its \( l^{th} \) level parent structure and compare it with the \( l^{th} \) level parent structure of all texels from the

\(^4\)We used an averaging filter.
sample set in the same location, i.e. we compare \( \mathbf{PS}_i(t)(m,n) \) with \( \mathbf{PS}_i(T^*_s)(m,n) \) for all \( i \). The best matching sample texel, \( T^*_s(m,n) \), is the sample with the most similar parent structure. Note that we have used \( i^* \) to denote the index of the sample texture map in which this most similar texel to \( t(m,n) \) was found. In reality this index differs for each input texel, so a more appropriate notation would be \( i^*(m,n) \). However, we use \( i^* \) for brevity.

The prior is then defined such that the high-frequency information inside the \( w \times w \) patch of the sought HR texture map that corresponds to texel \( t(m,n) \), is similar to the high-frequency information in the \( w \times w \) patch of \( G_0(T^*_s) \), in the same location. High-frequency information, in this context, is represented by the first order horizontal and vertical derivatives. In other words, we take the corresponding level zero gradients of each best matching texel, \( H_0(T^*_s)(p,q) \) and \( V_0(T^*_s)(p,q) \) (for all \( (p,q) \in \text{bin}(m,n) \)), as a prediction for the gradients of the sought HR texture map, in the same locations.

The gradient prediction algorithm that finds the predicted level zero horizontal, \( \hat{H}_0(T) \), and vertical, \( \hat{V}_0(T) \), gradient maps can be summarised as follows:

For each texel \( (p,q) \), do:

1. Find \( m = \lfloor \frac{p}{w} \rfloor \) and \( n = \lfloor \frac{q}{w} \rfloor \)
2. Find \( i^* = \arg\min_i \| \mathbf{PS}_i(t)(m,n), \mathbf{PS}_i(T^*_s)(m,n) \| \)
3. Copy \( H_0(T^*_s)(p,q) \) to \( \hat{H}_0(T)(p,q) \); and \( V_0(T^*_s)(p,q) \) to \( \hat{V}_0(T)(p,q) \).

Following [10], we use a weighted L2-Norm in the second step where the gradient components are given half as much weight as the Laplacian components and the weights are halved for each increase in the pyramid level.

Having formed the predicted gradient maps, the prior term is defined such that the gradients of the sought HR texture map are similar to these predicted values:

\[
-\ln[p(T)] = \frac{1}{\sigma_H^2} \sum_{p,q} [\mathcal{H}_0(T)(p,q) - \hat{H}_0(T)(p,q)]^2 + \\
\frac{1}{\sigma_V^2} \sum_{p,q} [\mathcal{V}_0(T)(p,q) - \hat{V}_0(T)(p,q)]^2
\] (6.13)

where we have assumed that the gradient prediction procedure has an error with an i.i.d Gaussian distribution with covariances \( \sigma_H^2 \) and \( \sigma_V^2 \) for horizontal and vertical gradients, respectively.

The final cost function to be optimised is the sum of the likelihood (Equation 6.8) and the prior (Equation 6.13) terms. Texture super-resolution is performed by optimising this cost function using the gradient descent algorithm.
6.2.2 Results

Figure 6.2 illustrates results of applying the texture super-resolution algorithm to low-resolution texture maps extracted from LR images. The magnification ratio in these images is 8. Note that in order to fit the images within the page, the texture maps were scaled to 40% of the original size. As a result, the improved resolution is not clearly visible. Therefore, we have also included a region of the texture maps (marked by a rectangle) at the original scale for better visual comparison in Figure 6.3.

It is clear from Figures 6.2 and 6.3 that the resolution of the LR texture map was enhanced by applying texture super-resolution. The high-frequency information injected into the texture maps seems visually acceptable and appropriate. The super-resolved (SR) face texture can be considered as a good estimated of the original HR texture.

Having estimated the high-resolution facial texture through the proposed 3-step framework, one can use the SR texture map together with the shape information estimated in step 1, $\alpha^*$, in order to render the face under arbitrary imaging conditions. Figure 6.4 illustrates this for 5 different poses of the same subject. In each row, column (a) shows the original HR image. A low-resolution image was obtained by down-sampling this image with a factor of 8.

Column (b) shows this LR image, enlarged by bilinear interpolation. Notice the amount of detail lost in the LR image. We super-resolve the facial texture by applying our proposed 3D-assisted framework to the LR input. The super-resolved texture was then used together with the estimated face shape in order to render the face with SR texture in the original pose. Column (c) shows the result. Comparing the images in column (c) with columns (a) and (b) shows that a considerable amount of the high-frequency detail which was lost in (b) is recovered and injected back into the texture by our approach. Finally, the super-resolved texture and the estimated facial shape were used to render the face in a normalised frontal pose, as illustrated in column (d). Note that all images in this figure were down-scaled to 50% of their original size in order to fit within the boundaries of the page. A full-scale version of these images is included in the Appendix for better visual comparison.

The results presented in this section show that the proposed framework is able to generalise the prior information learnt from a limited set of exemplar face images in order to super-resolve facial images of arbitrary imaging conditions. Note that the presented results were all generated using only a set of frontal images as the sample set. However, our proposed framework successfully uses the prior information learnt from these samples in a normalised texture domain in order to super-resolve facial images of non-frontal poses.

6.2.3 Further Improvements

The results presented in the previous section show that the proposed framework is capable of recovering relevant high-resolution information in the facial texture and successfully super-resolving face images under previously unseen imaging conditions. In this section we discuss some limitations of the proposed framework and present suggestions for potential further improvements.
6.2. Proposed Framework

Figure 6.2: Results of the proposed 3D-Assisted Facial Texture Super-Resolution for two samples. Each column illustrates from top to bottom: Original input image; LR texture map; and the super-resolved texture map.
Figure 6.3: Low-resolution vs. super-resolved texture at original size. Top row: Original LR texturemap with the region of interest marked, (scale=35%). Middle row: Region of interest in the low-resolution texture map (scale=100%). Bottom row: Region of interest in the super-resolved texture map (scale=100%).
Figure 6.4: Texture super-resolution in various poses and rendering under normalised imaging conditions. Each column shows (a): Original HR image; (b): (Bilinear interpolation of) the LR input; (c): Face rendered with SR texture in the original pose, (overlaid on the HR image); and (d): Face rendered with SR texture, in normalised frontal pose.
Model Fitting and Texture Extraction

The proposed 3-step framework effectively means that the face image registration in conventional face super-resolution will be replaced by model fitting and texture extraction. Thus the accuracy of these steps is of crucial importance. Although the LRF approach offers improved fitting performance for LR inputs, there are still areas for improvement.

A noticeable artefact can be spotted in the texture map shown in the right column of Figure 6.2, where a blue region is inserted in the texture map. This is due to misalignment of the 3DMM with the input LR image during the model fitting stage. Here, the fitted model is slightly larger than the input face. As a result, a part of the background is mapped to the LR texture map by mistake. Figure 6.5 shows a more extreme case of such an error.

![Figure 6.5: Artefact caused by model misalignment. (a) Shows the alignment of the 3DMM’s boundary (white line) with the boundary of the face after fitting. Notice the misalignment where the model’s boundary is slightly outside the face. (b) Shows the low-resolution texture map extracted from (a). Parts of the background are included in the texture map due to a small misalignment of the model and face boundaries.](image)

A possible solution would be to assign a measure of confidence to each part of the fitted model which describes the quality of the fit in the given local region. For instance, the pixel colour error can provide an indication of the fitting quality at the pixel level. In the regions of the face where the pixel colour error is low, one could assume, with high confidence, that the model fitting is accurate. On the other hand the regions with high pixel colour error are regions of likely misalignment between the model and the input image. Such a measure of confidence can then be used to limit texture extraction to areas where the fitting confidence is high. The 3DMM’s estimated texture can be used, instead of the extracted texture, in regions with a high chance of model misalignment.

Furthermore, one could take advantage of the prior knowledge that a human face is fairly symmetric, in a number of ways. Firstly, it could be used to improve the model fitting stage. We discuss the advantages of using facial symmetry in model fitting in more detail in the next Chapter. Another potential use of facial symmetry could be envisaged in the texture extraction process. For instance it could be used to detect artefacts such as the one in Figure 6.5 where the extracted texture on one side of the face is highly inconsistent with the texture extracted on the symmetric point of the face. Furthermore, when one side of the face is occluded, facial symmetry could be
used to extract texture from the symmetric points of the face instead of using the model’s estimated texture.

**Illumination Normalisation**

We mentioned in the previous section that information about scene illumination, as estimated by the model fitting process, can be utilised in order to de-illuminate the facial texture and extract the face albedo. In the present work, we have used a simple approach for de-illumination which involves inverting the Phong illumination model given the parameters estimated through model fitting. This simple approach relies heavily on the accuracy of the illumination model used and the parameters estimates provided by the fitting procedure. If the Phong model does not model the scene illumination sufficiently, or if the estimated parameters of the Phong illumination are not accurate enough, the simple de-illumination approach used here can cause artefacts in the de-illuminated texture. An example of this artefact is shown in Figure 6.6. In this case, there is an inconsistency between the illumination model used and the actual scene illumination conditions. In particular, the face is only lit by ambient light whereas the illumination model assumes both ambient and directional lights, thereby allowing extra degrees of freedom. This in turn results in inaccurate estimates for the illumination parameters. In particular, the contribution of the ambient light to the illuminated texture was under-estimated and the contribution of the directional light was over-estimated. As a result, the de-illuminated albedo is under-estimated in regions that the model assumes are lit by both light sources and is over-estimated in the regions that the model believes to be shadowed. Recall that, according to the illumination model, the areas in shadow are only illuminated by the ambient light. Thus, if the ambient light is under-estimated by the fitting process, de-illumination would result in an albedo value which is brighter than it should be. Also, if the directional light is over-estimated, the regions of the image which were lit by the directional light will have an unrealistically dark albedo after de-illumination.

![Figure 6.6: Artefact caused by error in the illuminations estimate. (a) Shows the input image. (b) Shows the extracted texture map. The region under the nose was identified by the model fitting algorithm to be obscured from the directional light (cast shadow). Facial albedo is over-estimated in this area.](image)

In addition to improving the estimate of the scene illumination, one could also consider using the available information to improve the de-illuminated texture estimation process. Again, facial symmetry is an important option to consider. For instance, one
could consider de-illuminating symmetric points simultaneously. This could potentially improve the de-illumination performance in areas where part of the face is in shadow while the symmetric part is lit by directional light, or when the texture on a particular point is lost due to specular highlights while the symmetric point is not affected by specular highlights.

**Texture Map Correction**

As mentioned previously, information about the accuracy of the fitted model, reflected by the pixel colour error, as well as symmetry of the face, can be used during the model fitting and texture extraction stage in order to improve the texture extraction. Alternatively, such information can be used in the texture domain to correct the artefacts after texture extraction.

We use the above-mentioned criteria to detect some of the texture map artefacts and correct them by replacing the extracted texture in the affected areas by the model’s estimated texture value. We define a texel error map by measuring the absolute error between the values of the extracted texture and the model’s estimated texture over all texels. This error map is then used to produce a confidence mask by comparing the error with a pre-defined threshold in each of the three colour channels. On this mask, areas with a high confidence (low texel error in all colour channels) are masked out while texels with low confidence (high texel error in any of the colour channels) are marked as potential artefacts; thus, subject to further processing.

We then define a symmetry error map by measuring the absolute difference between the extracted texture in symmetric texels. Again, all texels for which the error is higher than a given threshold are marked as potentially affected by artefacts.

By taking the overlap of these two masks and applying a series of morphological image processing operations\(^5\) to eliminate isolated small regions, we create a final mask which is then used for replacing the texture values. For all texels marked on this final mask, the extracted texture is replaced by the model’s estimated texture value. Figure 6.7 illustrates the masks and the corrected texture map for the artefact previously illustrated in Figure 6.5.

We used a threshold of 50 grey levels for both error types. However, as a result of the morphological image processing step, the final result shows a fair degree of robustness to the threshold value. We found that for a large range of threshold values the same simple procedure removes most of the misalignment artefact. Figure 6.7 shows the results of this procedure for a different sample using a number of different thresholds. A threshold too low can unnecessarily remove too much of the extracted texture and a threshold too high can leave parts of the artefact untouched. However, a large variation is tolerated before these effects become significant.

\(^5\)More specifically, we used a chain of operations consisting of image opening, closing, opening, and dilation operations, in that order.
6.2. Proposed Framework

Figure 6.7: Texture map correction for removing the artefact caused by model misalignment. (a) Initial LR texture map. (b) Confidence mask: shows texels for which the fitting confidence is low due to high pixel colour error. (c) Symmetry mismatch mask: shows the texels for which the absolute difference between the extracted texture for symmetric texels is high. (d) Combined mask: shows the overlap of the two masks. (e) The combined mask after morphological image processing. (f) Final LR texture map after removal of the artefact.

Figure 6.8: Texture map correction using different thresholds. Notice that the procedure used for removing the artefact caused by model misalignment has also removed specular reflections from the glasses and even (at low thresholds) parts of the glasses itself.
6.3 3D-Assisted Face Recognition in Low Resolution

We mentioned in Section 2.6 that two possible approaches can be envisaged when using a 3DMM for face recognition. One approach is to use the model’s estimated parameters directly. We used this approach in Chapter 5. An alternative approach, which we refer to as 3D-assisted face recognition, involves using the 3DMM in order to normalise the effects of extrinsic factors such as pose, illumination, and resolution in a 2D probe image. Recognition can then be performed using a conventional 2D face recognition engine, with the resulting normalised probe image as input.

The original facial texture, where available, can be used in the 3D-assisted approach by mapping the facial texture from the original image to the normalised image. However, in the case of an LR probe image, the facial texture would normally be interpolated in order to render the probe face in normalised size. This produces an image with blurred texture which is unsuitable for recognition. To address this problem, the 3D-assisted texture super-resolution framework, proposed in the previous section, can be used in order to enhance the facial texture prior to rendering the normalised probe image.

In order to demonstrate the feasibility of such an approach, we present a face verification experiment using the XM2VTS dataset. Similar to the experiment presented in Section 5.4, the first configuration of the Lausanne protocol is used in this experiment.

We begin by fitting the 3DMM, using the MFF algorithm, to the HR training and evaluation images and rendering the faces using a pre-defined set of rendering parameters corresponding to normalised imaging conditions. The resulting normalised images are used for training and setting the operating threshold of a 2D face verification system.

We then generate the LR probe images by down-sampling the test set by a factor of 8, and applying our 3D-assisted facial texture super-resolution framework to render the LR images using the super-resolved texture and the same rendering parameters which were used to render the training and evaluation samples. We used the HR client training images to build the texture sample set used in the texture super-resolution step.

The resulting super-resolved probe samples are then classified by the face verification engine into clients and imposters, similar to the experiment presented in Section 5.4. For comparison, we also present results in the case where the original LR texture is used for rendering the normalised probe images, i.e. without texture super-resolution.

Any conventional 2D face recognition engine can be used within this framework. We use a recognition system similar to the one proposed by Shan et al. [95]. An ensemble of classifiers is built by dividing the image into 36 rectangular regions. Spatial histograms of uniform Local Binary Patterns (LBP) [3] are used in each region as feature vectors which are then projected into LDA space. A sum rule is then used to combine the similarity scores (normalized correlation) corresponding to each of the 36 regions.

We process each colour channel separately in the RGB colour space. The final similarity scores for comparing two colour images is obtained by fusing the scores of all three colour channels using the sum rule. Similar to the experiment in Section 5.4, an access score is generated for each claim by fusing access scores corresponding to comparison of the probe image with each of the 3 gallery samples of the claimed identity. Here, we use
the \textit{min} rule in the evaluation stage while using the \textit{max} rule in the test stage. Finally, Nearest Neighbor (NN) is employed for classification.

Table 6.1 compares the Half-Total Error Rates (HTER) for the two experiments where LR or SR texture maps were used. It is apparent that the proposed framework improves the verification performance significantly by enhancing the facial texture in the probe images.

<table>
<thead>
<tr>
<th></th>
<th>Front</th>
<th>Left</th>
<th>Right</th>
<th>Up</th>
<th>Down</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR texture</td>
<td>16.16%</td>
<td>22.19%</td>
<td>22.37%</td>
<td>26.50%</td>
<td>18.45%</td>
<td>21.13%</td>
</tr>
<tr>
<td>SR texture</td>
<td>8.49%</td>
<td>12.36%</td>
<td>11.97%</td>
<td>12.36%</td>
<td>10.62%</td>
<td>11.16%</td>
</tr>
</tbody>
</table>

Comparing the results in Table 6.1 with those presented in Table 5.7 for the experiment which used the model parameters for recognition would initially suggest that the model parameters yield better performance. However, this is not a generalisable conclusion. One obvious reason is that the framework used for the present experiment can use any conventional 2D face recognition system. Thus, the final recognition performance relies heavily on the specific recognition engine used. A more powerful 2D recognition method might be able to achieve better results using the same super-resolved images. Further investigation into an optimal 2D face recognition engine is outside the scope of this work. Nevertheless, the fact that the verification performance is significantly improved using the proposed 3D-assisted facial texture super-resolution method confirms that the additional HR information injected into the LR probe image is beneficial for recognition.

In addition to using a more powerful recognition engine, the performance reported in Table 6.1 can potentially be improved in a number of other ways. We presented some suggestions for improving the performance of the texture super-resolution algorithm in Section 6.2.3. The texture map correction procedure of Section 6.2.3 was not used in this experiment. Using this procedure to enhance the super-resolution performance could in turn improve the verification performance.

Moreover, as mentioned earlier, we have used the RGB colour space for this experiment and fused similarity scores over the three channels. However, this may not be an optimal approach as suggested by previous studies which have investigated the use of various different colour spaces (for example, see [91]).

Another way to potentially improve the performance of the proposed 3D-assisted framework is to differentiate between \textit{estimated} and \textit{extracted} texture. Recall that we extract texture from the image where available, but in the occluded areas where the original texture is not available, we use texture values estimated by the model. Any small mismatch between the estimated and extracted texture values can produce artificial edges in the texture. Such edges may be falsely interpreted as facial features by the recognition engine. However, since we know the areas for which texture was extracted, it is straightforward to detect such edges and discard the false features in such areas. For instance, in the face recognition system described here, one could avoid including LBP features along these false edges in the spatial histograms.

Furthermore, considering that the texture estimated by the model is generally smoother, and potentially less accurate, than the original facial texture, one could chose to limit
the comparison to areas of the face for which original texture is available, and completely exclude all regions with estimated texture from the comparison. Alternatively, one could weight the contribution of different regions of the face to the final similarity score differently, based on whether they are composed of “extracted” or “estimated” texture. For instance, in the system used here, we sum the similarity scores in 36 rectangular regions to obtain an overall similarity score. Instead, one could use a weighted sum rule where the weight of each region is proportional to the ratio of pixels with “extracted” texture to the total number of pixels within the given region.

6.4 Discussion

We presented a framework for facial texture super resolution independently of the facial shape, subject pose, and scene illumination. Equation 6.1 gives the foundation for our framework. However, finding the optimal HR texture from the aforementioned equation is not a trivial task. Consequently, we made a number of simplifying to arrive at the 3-step framework given in Section 6.2.

Although in our experimental evaluation in the previous sections we have used a specific 3DMM (the CVSSP-3DMM) and a specific fitting algorithm (MFF for HR samples and LRF for the LR inputs), the basic framework is independent of these individual components. In [78], we showed that the same framework can yield impressive results even using a less descriptive model and a much simpler fitting method. The model used in [78] was an earlier version of the CVSSP-3DMM, which was build using only 69 training faces and only includes grey scale texture. Also, the SNO algorithm was used for fitting this model to the LR input images. Figure 6.9 shows some of the results and compares our method with Baker and Kanade’s reconstruction method [10] applied directly in the image domain. Both methods use the same set of images for their sample set.

Figure 6.9 shows that, even when using a less discriminative 3DMM and a simpler fitting algorithm, our 3-step framework can still yield results which are visually comparable to the results of applying the reconstruction framework directly in the image domain, with the added benefit that our framework enables super-resolution of previously unseen poses and illuminations which is not possible with the reconstruction algorithm.

In deriving the 3-step framework, we replaced the distribution $p(T|\mu^*, \rho^*, L)$ with $p(T|t)$ (Equation 6.5). We then made an implicit assumption that the same observation model used in the image domain can also be used in our normalised texture domain and proceeded to super-resolve the texture map in a MAP estimation framework using this observation model to formulate the likelihood term. In doing so, we have effectively neglected the difference between the texture maps and regular 2D images.

However, a closer look at the procedure through which we generate our normalised texture maps (Section 2.5), reveals that the aforementioned assumption is not necessarily correct. That is, the observation model used in the image domain, which describes an LR image in terms of low-pass filtering and down-sampling an HR image does not describe the formation process of an LR texture map. An LR texture map is the result
Figure 6.9: Super-resolution of frontal faces. Top row: (Bilinear interpolation of ) the LR images; second row: Baker and Kanade's reconstruction method; third row: Our method; fourth row: Original HR images.
of texture extraction from an LR image which in turn is the result of low-pass filtering and down-sampling an HR image. An HR texture map, on the other hand, is the result of texture extraction from an HR image. Figure 6.10 illustrates this difference. This suggests that a simple observation model such as the one in Equation 6.7 is an unrealistic model for describing a low-resolution texture map in terms of a high-resolution one. At the least, the low-pass filter kernel needs to be operating on an irregular grid and potentially have a spatially varying response which in turn depends on the facial shape and the subject’s pose.

In [77], we presented an alternative approach which avoids making the above-mentioned “unrealistic” assumption. More specifically, instead of replacing the distribution of \( p(T|\mu^*, \rho^*, L) \) with \( p(T|t) \), we formulate the problem as:

\[
T^* = \arg\max_T \{ p(T|\mu^*, \rho^*, L) \} \\
= \arg\max_T \{ p(L|T, \mu^*, \rho^*)p(T) \} \\
= \arg\min_T \{ -\ln p(L|T, \mu^*, \rho^*) - \ln p(T) \} \tag{6.14}
\]

In this alternative approach, instead of using the optimal model and rendering parameters, \( \mu^* \) and \( \rho^* \), to generate an intermediate LR texture map, \( t \), we use them to formulate a generative model which describes the formation of an LR image, given the sought HR texture map, schematically illustrated by path (e) in Figure 6.10. This generative model is then used to define an image-based likelihood term in the above MAP estimation framework for texture super-resolution.

Figure 6.10: Relationships between the image and texture domains. Red arrows show texture extraction paths while black arrows show image formation paths. Valid paths are shown by solid lines. In particular: (a) and (b) show texture extraction from HR and LR images, respectively. (c) Shows an image-domain observation model (e.g. Equation 3.3, used in image super-resolution). (d) Shows the “unrealistic” texture-domain observation model used in our 3-step framework. (e) Is the texture-to-image generative model used in our IBL approach.
We refer to this alternative approach as the image-based likelihood (IBL) approach. In the following, we briefly describe this approach and compare it with the 3-step framework proposed in Section 6.2.

The sought HR texture map represents shape- and pose-normalised texture of the face. Hence, in order to generate an image from such a texture map, the image formation process includes projecting all visible texels to the image plane according to the shape and pose information, and integrating the texture values within each pixel of the image plane to obtain the final pixel value. In order to project any point from the texture map to the image plane, the equivalent point on the 3D surface is found and projected to the image through a pinhole camera model.

In order to project a texel to the image plane, we project its 4 corners and obtain a quadrangle on this plane. We define each pixel \((x, y)\) of the image as “affected” by texel \((p, q)\)” if the area of the overlap between pixel \((x, y)\) and texel \((p, q)\), when projected to the image plane, is non-zero. Each pixel value is then defined in our generative model as the weighted sum of the texel values which affect it. The weights define the contribution of each texel to the pixel value:

\[
L(x, y) = \sum_{p,q} \frac{W(x, y; p, q; \alpha^*, \rho^*)}{\sum_{p,q} W(x, y; p, q; \alpha^*, \rho^*)} T(p, q) + \eta_p(x, y) \tag{6.15}
\]

where \((p, q)\) indexes the texels of the texture map \(T\), \((x, y)\) indexes the pixels of the image \(L\), \(W\) is the weight that determines the contribution of texel \(T(p, q)\) to the value of pixel \(L(x, y)\), and \(\eta_p\) is an additive pixel noise term. The weight \(W(x, y; p, q; \alpha^*, \rho^*)\) is defined as the fraction of the area of the projected texel that lies within pixel \((x, y)\) on the image plane.

Note that projecting any point of the texture map to the image plane requires information about the 3D shape of the model in order to find the equivalent point on the 3D face surface as well as pose and rendering parameters in order to project the point from the 3D surface to the image plane. Hence, \(W\) in Equation 6.15 is a function of \(\alpha^*\) and \(\rho^*\).

Using the generative model in (6.15) and assuming that the pixel noise \(\eta_p\) has an i.i.d Gaussian distribution with covariance \(\sigma_{\eta_p}^2\), the (negative log of the) image-based likelihood can be defined as:

\[
-\ln p(L|\mu^*, \rho^*, T) = \frac{1}{\sigma_{\eta_p}^2} \sum_{x,y} \left[ L(x, y) - \sum_{p,q} \frac{W(x, y; p, q; \mu^*, \rho^*)}{\sum_{p,q} W(x, y; p, q; \mu^*, \rho^*)} T(p, q) \right]^2 \tag{6.16}
\]

Finally, texture super-resolution is performed in a MAP estimation framework with the likelihood given by Equation 6.16 and the prior term given by Equation 6.13.

The IBL approach avoids making the aforementioned “unrealistic” assumption and improves the performance of our 3D-assisted facial texture super-resolution framework.
However, this improvement is not large enough to undermine the use of our previous 3-step approach.

Figure 6.11 shows some examples of texture super-resolution using the two approaches. Note that these results were obtained using an earlier version of the CVSSP-3DMM which only included 69 training samples, and that the SNO algorithm was used for fitting this model to the images. Furthermore, grey-scale images were used in these experiments.

The results shown in Figure 6.11 show that both methods produce visually similar results, and the perceived resolution enhancement is similar using both methods.

Additionally, we compare the ability of the two proposed methods in providing discriminative high-resolution information useful for recognition by performing a low-resolution face recognition experiment. Table 6.2 compares the rank-1 identification rates achieved with the two approaches. Here, the LR probe images are 8 times smaller than the HR gallery images in each dimension. We have used frontal images of 140 clients from the XM2VTS dataset in this identification experiment. All 8 frontal shots of the imposter images (95 subjects) were used as the sample set for texture super-resolution. The face recognition engine used in this experiment is similar to the one used in the previous experiment in Section 6.3. The only difference is that here we have divided the images into 100 regions instead of 36.

As shown in Table 6.2, the high identification rate in HR decreases considerably when LR probe images are used. This rate is improved significantly by both texture super-resolution approaches proposed in this chapter. However, the performance of the image-based likelihood approach is only marginally superior to that of the 3-step approach.
Table 6.2: Comparison of Rank-1 Identification rates for HR inputs, LR inputs enlarged by bilinear interpolation, LR inputs super-resolved using Baker and Kanade’s Face hallucination approach [8] in the image domain, LR inputs super-resolved by the 3-Step framework proposed in Section 6.2, and the alternative Image-Based Likelihood (IBL) approach presented in Section 6.4

<table>
<thead>
<tr>
<th>Method</th>
<th>Identification Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td>99.28%</td>
</tr>
<tr>
<td>LR + Bilinear Interpolation</td>
<td>78.57%</td>
</tr>
<tr>
<td>LR + Baker-Kanade</td>
<td>96.43%</td>
</tr>
<tr>
<td>LR + Texture Super-Resolution (3-Step)</td>
<td>93.57%</td>
</tr>
<tr>
<td>LR + Texture Super-Resolution (IBL)</td>
<td>95%</td>
</tr>
</tbody>
</table>

This suggests that although the 3-step approach was derived by making an “unrealistic” assumption about the texture generation process, this assumption is in practice useful and does not have a significant negative effect on the performance.

Table 6.2 also shows the recognition results for frontal faces when super-resolution is performed in the image domain using Baker and Kanade’s face hallucination approach. Note that this approach yields slightly better results than both of our proposed approaches. However, our proposed approaches have the ability to super-resolve facial images with arbitrary pose whereas the image-domain approach can only super-resolve frontal faces. Nevertheless, the increased performance obtained with our approaches shows that the high-frequency information added to the image using our approaches is beneficial for recognition and is comparable to the image-based face hallucination approach.

6.5 Conclusions

This chapter presented a framework for unconstrained single frame facial texture super-resolution using 3D information. We derived a 3-step framework which includes fitting a 3DMM to the LR image at hand, followed by mapping facial texture from the LR input image to a normalised domain in which the effects of facial shape as well as extrinsic factors such as pose and illumination are normalised. This texture is then super-resolved using an example-based super-resolution method. We also provided suggestions for improving the basic framework.

The proposed pipeline offers a generic framework which is independent of the individual components, thus providing a powerful practical tool for processing LR facial images. It was demonstrated through exemplar results that the framework is able to inject visually acceptable high-resolution detail into the given LR texture. We also provided an example face recognition experiment in order to show how the proposed framework can be used for recognising faces in low-resolution. The results of this experiment confirmed that the added high-resolution detail is not only visually acceptable, but is also beneficial for recognition.

Finally, we discussed that the proposed 3-step framework makes an unrealistic assumption about the texture generation process and presented an alternative approach (IBL)
which avoids such an assumption. Evaluation of this alternative approach revealed that the performance of the basic 3-step framework is only slightly compromised due to the aforementioned unrealistic assumption and that the negative effects are negligible. Both approaches yield similar results both visually and in terms of the amount of discriminative information they provide for face recognition. However, from a practical point of view, the 3-step framework has the advantage of being completely modularised in the sense that it is able to make use of stand-alone fitting and super-resolution modules while the IBL approach uses a tailored observation model in the super-resolution step which in turn depends on the model fitting results.
Chapter 7

Conclusions and Future Work

This work investigated the use of 3D information for unconstrained face recognition using 2D low-resolution (LR) images. More specifically, we consider a face recognition scenario where subjects are enrolled using 2D high-resolution (HR) images, but the images presented to the system for recognition are low-resolution. Unconstrained recognition, in this context, refers to a non-cooperative scenario where the probe images are not assumed to be taken under constrained conditions. That is, the probe image may have a different pose to the gallery images and may be taken under illumination conditions which differ from that of the gallery images. The only constraint assumed in this work is for the probe face to have a neutral facial expression.

We investigated different approaches to the problem of LR face recognition in Chapter 3 and categorised these approaches into four main groups. We argued that most of the current approaches are unable to handle variations in the unconstrained probe case without making rather impractical assumptions, for example about availability of sufficient training data in different poses and illuminations.

We used a 3D Morphable Face Model (3DMM) for modelling and extracting 3D information from a 2D image. Inspired by the super-resolution literature, a novel method was proposed for fitting a 3DMM to LR images. Our approach uses an LR imaging model to describe the formation of an LR image given the parameters of a 3DMM and a set of rendering parameters. We showed that our proposed Low Resolution Fitting (LRF) algorithm is able to extract high-resolution 3D information from low-resolution 2D images more reliably than the conventional fitting algorithms.

The 3D information extracted from an LR image was used in Chapter 5 for face recognition where the 3DMM’s parameters were used directly. We used two alternative methods for this purpose. The first method was the approach proposed by Blanz et al. [20]. In this approach, the shape and texture parameters of a face are rescaled by their respective standard deviations and stacked together in an identity vector. Comparison of two faces is then performed by finding the similarity between their respective identity vectors. Normalised Correlation (NC) was used as the measure of similarity. The second method was a novel LDA-based approach, proposed in Section 5.4. In this approach a separate LDA transform is trained for each of the global and local shape and texture parameter vectors corresponding to a face. Comparison of two faces is...
then performed by projecting all parameter vectors corresponding to each face to their respective LDA spaces and measuring similarity (NC) of the projected samples. The final similarity measure for two given faces is a weighted sum of the individual scores corresponding to each of the parameter vector pairs.

We presented face identification experiments in low-resolution using the PIE dataset (Section 5.3) with a range of variations in pose and/or illumination of the probe images. It was shown that the 3D information extracted using our proposed LRF approach can yield impressive face identification performance across a wide range of resolutions, poses, and illumination conditions and that in all LR cases our proposed LRF fitting outperforms the conventional MFF algorithm.

We also presented a face verification experiment using the XM2VTS dataset (Section 5.4) and our proposed LDA-based face recognition method. Again, it was shown that our method can yield a high verification performance in low-resolution images over different poses.

Finally, we presented a framework for using 3D information for unconstrained facial texture super-resolution in Chapter 6. It was shown that, using our framework, facial texture of various poses can be super-resolved using a sample set consisting of frontal faces only. Also, the framework can potentially be used for normalising the effects of varying illumination conditions, although our attempts for illumination normalisation were not always successful due to the simple illumination model used in the model fitting stage; hence, accurately estimating and normalising the illumination remains an open problem.

7.1 Future work

The 3DMM has proved to be a powerful tool for facial analysis and modelling. Due to its ability to separate extrinsic and intrinsic sources of variation in a single 2D image, it can be regarded as a potential solution for the long standing problem of truly unconstrained face recognition. The author of this thesis believes that the extension of 3DMM usage to the LR case, presented in this work, is a step in this direction. However, this is not a solved problem yet and there are many issues to be resolved.

7.1.1 Efficiency

Arguably, one of the biggest challenges in fitting a 3DMM to a 2D image is the efficiency of the fitting algorithm. Different approaches have been considered in the literature for improving the fitting efficiency. For instance, by extending the Inverse Compositional Image Alignment [11] algorithm, Romdhani et al. [89] proposed an efficient fitting algorithm for fitting a 3DMM. Although this algorithm improves the fitting efficiency, it is not able to handle directed light (See [88] for a more detailed discussion of the algorithm and its shortcomings).

The fitting efficiency can be improved by limiting the number of PCA coefficients fitted in each stage or evaluating the cost functions and their derivatives over a subset of the vertices. However, such approximations would sacrifice the accuracy of model fitting.
Another strategy for speeding up the fitting could be to use a multi-resolution fitting strategy where a low-resolution model is first fitted to the input image, or a down-scaled version of it, and the results are used to initialise higher levels of resolution. Due to less computational cost, fitting a low-resolution model can be more efficient than a high-resolution model. Then, the result of fitting this model can be used to initialise the HR model near the optimum which in turn will mean that the algorithm will eventually converge with fewer iterations.

Finally, since the fitting algorithm includes numerous vector calculations which can be performed in parallel, implementing the algorithm on a GPU can potentially be much more efficient than using a conventional CPU.

### 7.1.2 Multi-Frame Model Fitting

This work focused on inferring HR information from a single LR image. In general, if multiple LR observations are available, for example in a CCTV footage, fusing the information across the multiple LR observations can considerably improve the accuracy of the estimated HR information. An example of this is multi-frame super-resolution algorithms which use multiple LR images in order to reconstruct an HR image and generally have better performance than their single-frame counterparts. We believe that such improvements can also be gained by simultaneously fitting the 3DMM on multiple LR images.

The LRF algorithm proposed in Chapter 4 can be extended to fit the 3DMM simultaneously on multiple low-resolution images. In a simple case, one can ignore all correspondences between the different inputs except the identity. This is effectively equivalent to assuming that the different input images are taken from the same individual independently and in potentially different imaging conditions. In this case, all inputs share the same model parameters, $\mu = \{\alpha, \beta\}$, while each input image has an independent set of corresponding rendering parameters, $\rho_j = \{\tau_j, \gamma_j\}$. Here, $\tau_j$ and $\gamma_j$ reflect the specific registration and illumination conditions of the $j^{th}$ input image, respectively. Then the single-frame LR observation model of Equation 4.21 (Section 4.3.3) can be expressed for the $j^{th}$ input image as:

$$I_{j}^{\text{model}}(m) = \frac{1}{A_j} \sum_{k \in K_j(m)} W_j(k, m) \hat{t}_{j,k}^C$$

where $I_{j}^{\text{model}}(m)$ is the model’s estimate for pixel $m$ of the $j^{th}$ input image and the rest of the symbols are as defined in Equation 4.21 with the only difference being that here, they correspond to the $j^{th}$ input image. In particular, $\hat{t}_{j,k}^C$ is the texture of the $k^{th}$ triangle of the model, after being illuminated and colour transformed using the illumination parameters $\gamma_j$, and $W_j(k, m)$ is the area of overlap between pixel $m$ of the $j^{th}$ input image and the $k^{th}$ triangle of the model after being projected to the image plane of $I_j$ using the registration parameters $\tau_j$. 

The fitting algorithm can then be extended in order to fit the model simultaneously to multiple images by summing each image-based cost function over all input images. For example, the pixel colour cost function can be written as:

\[
E_{\text{LRF}}^{c} = \sum_{j=1}^{N_{\text{inp}}} E_{\text{LRF}}^{c,j} = \sum_{j=1}^{N_{\text{inp}}} \sum_{m} \| I_{j}^{\text{input}}(m) - I_{j}^{\text{model}}(m) \|^2
\]

(7.2)

where \( N_{\text{inp}} \) is the total number of LR input images and \( E_{\text{LRF}}^{c,j} \) is the pixel colour cost function for the \( j^{\text{th}} \) input image, equivalent to Equation 4.24 for the single-frame case. The Anchor cost and Edge costs can also be extended in a similar way by summing over the multiple inputs for the multi-frame case.

In the above, it was assumed that the multiple LR inputs are independent. However, it may be possible to assume some correspondence between the different input images and exploit such correspondences in the fitting algorithm. For instance if the different images are successive frames of a CCTV video sequence, one can safely assume that the subject’s pose and the scene’s illumination are unlikely to change significantly between two consecutive frames, in other words, it can be assumed that \( \rho_{j} \simeq \rho_{j-1} \). Accordingly, a cross-image cost function can be added to the set of cost functions to incorporate such knowledge in the fitting framework. For example, assuming the variations in the rendering parameters have an \( i.i.d \) Gaussian distribution, this cost function can be expressed as:

\[
\sum_{j=2}^{N_{\text{inp}}} \| \rho_{j} - \rho_{j-1} \|^2
\]

(7.3)

Alternatively, if one can measure or predict the inter-frame variations, more intelligent cost functions could possibly be formulated.

A further extension could take into account that a person’s face does not always have the exact same shape and texture. Accordingly, one could consider a limited amount of variation in the facial shape and texture of the model for each input image to account for mild variations in facial characteristic due to e.g. moderate expressions or change of skin tone over time.

In addition to fusing the information available from multiple LR observations during the fitting process, this information could also be used to enhance the estimate of high-resolution facial texture obtained with the 3D-assisted facial texture super-resolution framework proposed in chapter 6. The only difference here would be to extend the super-resolution step to multi-frame super-resolution [10].

### 7.1.3 Illumination Estimation

Illumination information is recovered in the current work by using the Phong reflectance model with only a single directional light besides the ambient light. The Phong model only provides a rough estimate of the reflectance properties of the human skin. Using a
reflectance model which can model properties of the human skin more accurately and an illumination model which can model a wider range of environments could improve the illumination estimate. However, such extensions could considerably increase the computational cost of the fitting algorithm.

A different approach in attempting to improve illumination estimation is to use prior knowledge about human faces and expected lights. The light regularisation cost function included in our work (Section 2.4.2) is an example of attempting to improve illumination estimation without incurring the extra computational costs associated with using more sophisticated illumination models.

Prior knowledge about symmetry in a human face can also be useful in improving the illumination estimate. The proportion of the albedo and shading in the intensity of a pixel is ambiguous during the fitting process. An incorrect proportion will affect the fitting quality, in particular for input images under strong point light illumination. The injection of any prior knowledge pertaining to the general human facial texture into the illumination estimation process is likely to decrease the inherent ambiguity. Although not perfectly symmetric, a human face exhibits a fair amount of symmetry. Such knowledge can be incorporated into the fitting process in the form of an extra constraint on the estimated facial albedo values:

\[
E_{sym} = \frac{N_v}{2} \sum_{i=1}^{N_v} \| t_{r,i} - t_{l,i} \|^2
\]

(7.4)

where \( t_{r,i} \) and \( t_{l,i} \) are the RGB albedos of the \( i^{th} \) model vertex from the right side of the face and its symmetric vertex on the left side, respectively. We have applied this idea to the case of fitting the 3DMM on HR images using the MFF framework and presented some initial results in [50]. It is shown that this constraint can improve the illumination estimate. Figure 7.1 compares some sample fitting results with and without the facial symmetry constraint of Equation 7.4.

7.1.4 Automation

Currently, the fitting is initialised by manually labelled landmarks. In order to achieve a fully automated approach, it is required that the landmarks are detected automatically. To this end, one option is to use feature detectors which are able to detect certain landmark points (e.g. eyes, tip of the nose, etc.) on the face. However, only a few points can be accurately detected in a face, and landmarks detection in presence of pose and light variation can be a challenge. Moreover, detection of landmarks is particularly challenging when it comes to low-resolution images.

Another option is to use a 2D Active Appearance Model (AAM). An AAM can be initialised with only a few feature points (e.g. only eye centres), or even with only the bounding box of the face. It has been shown that using multilinear subspace analysis, a tensor-based AAM can be built to cope with a reasonable amount of variation in poses and illuminations. Moreover, methods of fitting AAMs to LR images have been proposed in the literature ([33] and [68]). It is not clear whether these methods provide
Figure 7.1: Fitting results on HR images using the MFF algorithm. The first column shows the original images from the PIE dataset with side illumination, the second and third columns show the fitting results without and with the facial symmetry cost, respectively.

sufficient accuracy for face recognition in low-resolution, but they can possibly provide sufficient accuracy for initialising the 3DMM fitting.

7.2 A Plea to the Community

I intentionally started this thesis with a rather unusual example of identity fraud: a “legitimate” one! The intention is to signify the fact that despite all the merits of developing accurate and reliable biometric person identification systems for legitimate purposes, there is always the danger that technology in the wrong hands can lead to disasters.

The Romans of the story had no means of verifying the identity of the claimants. Had they been equipped with modern day technology, they would not have even felt the need to offer leniency to the defeated army of rebel slaves. Although this particular event never actually happened and is only a Hollywood dramatisation, it is not too difficult to think of real-life modern-day examples where reliable biometric systems can be misused if fallen into the wrong hands.

A quick glance at today’s international political atmosphere is enough to realise that it is littered with oppressing regimes who would love to crack down on their rivals, oppositions, human rights activists, and others, using advanced biometric technology to identify and track them.

Therefore, alongside advancing the science and technology, it is a moral responsibility for us, researchers in the biometrics research community, to ponder upon possible ways of preventing misuse of such advanced and powerful technology.

Let us not help the tyrants and oppressors of our times!
Appendix A

Result of 3D-Assisted Facial Texture Super-Resolution

The following figures illustrate the images in Figure 6.4 at 100\% scale for better visual comparison.
Appendix A. Result of 3D-Assisted Facial Texture Super-Resolution

Figure A.1: Texture super-resolution in “frontal” pose and rendering under normalised imaging conditions. (a): Original HR image; (b): (Bilinear interpolation of) the LR input; (c): Face rendered with SR texture in the original pose, (overlaid on the HR image); and (d): Face rendered with SR texture, in normalised frontal pose.
Figure A.2: Texture super-resolution in “left” pose and rendering under normalised imaging conditions. (a): Original HR image; (b): (Bilinear interpolation of) the LR input; (c): Face rendered with SR texture in the original pose, (overlaid on the HR image); and (d): Face rendered with SR texture, in normalised frontal pose.
Figure A.3: Texture super-resolution in “right” pose and rendering under normalised imaging conditions. (a): Original HR image; (b): (Bilinear interpolation of) the LR input; (c): Face rendered with SR texture in the original pose, (overlaid on the HR image); and (d): Face rendered with SR texture, in normalised frontal pose.
Figure A.4: Texture super-resolution in “up” pose and rendering under normalised imaging conditions. (a): Original HR image; (b): (Bilinear interpolation of) the LR input; (c): Face rendered with SR texture in the original pose, (overlaid on the HR image); and (d): Face rendered with SR texture, in normalised frontal pose.
Figure A.5: Texture super-resolution in “down” pose and rendering under normalised imaging conditions. (a): Original HR image; (b): (Bilinear interpolation of) the LR input; (c): Face rendered with SR texture in the original pose, (overlaid on the HR image); and (d): Face rendered with SR texture, in normalised frontal pose.
Bibliography


