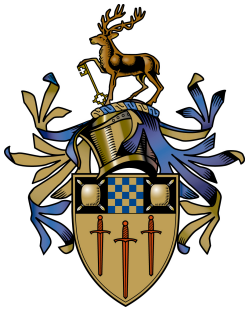
High-detail temporally consistent 3D capture of facial performance

Martin Klaudiny

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

© Martin Klaudiny 2013

Summary

Capturing a realistic digital copy of a facial performance has high importance for film and television production. This allows high-quality replay of the performance under different conditions such as a new illumination or viewpoint. The model of performance can be altered by space-time editing or can be used for building and driving a facial animation rig. This thesis presents a novel system to capture high-detail 4D models of facial performances. A geometric model without appearance is reconstructed from videos of an actor's face recorded from multiple views in a controlled studio environment. The focus is on achieving temporal consistency and a high level of detail of the 4D performance model which are crucial aspects for the use in film production.

A baseline method for dense surface tracking in multi-view image sequences is investigated for facial performance capture. Evaluation shows limitations of previous sequential methods which provide accurate temporal alignment only for faces with a painted random pattern. A novel robust sequential tracking is proposed to handle weak skin texture and rapid non-rigid facial motions. However, gradual accumulation of frame-to-frame alignment errors still results in significant drift of the tracked mesh. A non-sequential tracking framework is introduced which processes an input sequence according to a tree derived from a measure of dissimilarity between all pairs of frames. A novel cluster tree enables balancing between sequential drift and non-sequential jump artefacts. Comprehensive evaluation shows temporally consistent mesh sequences with very little drift for highly dynamic facial performances. Improvements are also demonstrated on whole-body performances and cloth deformation.

Photometric stereo with colour lights is used for capturing pore-level skin detail. An original error analysis of the technique is conducted for image noise and calibration errors. The proposed markerless capture system for facial performances combines photometric stereo with non-sequential surface tracking based on the cluster tree. A practical capture setup is constructed from standard video equipment without active illumination or high-speed recording. Errors in the photometric normals are corrected using the temporally aligned mesh sequence. The resulting 3D models enhanced by the normal maps capture fine skin dynamics such as skin wrinkling. High-quality temporal consistency of the models is also demonstrated with minimal drift in comparison to the previous approaches. Qualitative and quantitative comparison with the best state-of-the-art system shows comparable results.

Key words: facial performance capture, dense motion capture, non-sequential surface tracking, photometric stereo with colour lights

Email: m.klaudiny@surrey.ac.uk

WWW: http://www.surrey.ac.uk/cvssp/people/phd_students/martin_klaudiny/

Acknowledgements

I would like to thank my principal supervisor Prof. Adrian Hilton for introducing me to the topic of 3D facial capture and giving me an opportunity to work on it. His advise, guidance and 'bigger-picture' insights kept my research focused and made this thesis possible. I am also grateful to my co-supervisors Dr. James Edge and Dr. Eng-Jon Ong for discussions and feedback over the years.

Thanks also goes to the present and past members of Visual Media Group and the whole CVSSP for creating nice and friendly environment to work in. I would like to highlight a number of individuals who helped my research and shared the burden of PhD with me. I start with Dr. Chris Budd who has been a long-standing officemate and a research collaborator. I am indebted to Ashish Gupta, Alaleh Rashidnasab, Marco Volino and Cemre Zor for enduring long capture sessions as test actors. These sessions often required extra pairs of hands which were kindly provided by Dan Casas Guix, Dr. Evren Imre, Stuart James, Dr. Zdenek Kalal, Dr. Nataliya Nadtoka, Dr. Muhammad Awais Rana and Margara Tejera Padilla. I appreciate that they performed boring and repetitive tasks with little explanation what it is good for.

Finally, special thanks goes to my family and friends back home for their support and encouragement. They had always higher regard for what I have been doing than what it rightfully deserves.

Contents

1 Introduction **1**

1.1 Motivation 2

1.2 Objective 5

1.3 Thesis outline 6

1.4 Contributions 8

1.5 Publications 10

2 Related work **11**

2.1 Facial performance capture 11

2.1.1 Realism of the 4D performance model 11

2.1.2 Temporal consistency of the 4D performance model 23

2.2 Photometric stereo 33

2.2.1 Photometric stereo with colour lights 34

2.2.2 Photometric stereo with gradient illumination 35

2.2.3 Error analysis 36

2.3 Dense motion capture 36

2.3.1 Frame-to-frame scene flow 37

2.3.2 Sequential surface tracking 37

2.3.3 Non-sequential surface tracking 39

2.4 Conclusion 40

3 Geometric detail capture **42**

3.1 Photometric stereo with white lights 43

3.2 Photometric stereo with colour lights 45

3.3 Calibration 47

3.4 Error analysis . . . 49
3.4.1 Error in the illumination matrix . . . 49
3.4.2 Error in the interaction matrix . . . 52
3.4.3 Image noise . . . 53
3.4.4 Experiments . . . 54
3.4.5 Discussion . . . 60
3.5 Evaluation . . . 61
3.5.1 Reconstruction of facial performance . . . 63
3.5.2 Comparison to photometric stereo with white lights . . . 64
3.5.3 Effect of facial make-up . . . 65
3.6 Conclusion . . . 65

4 Baseline sequential surface tracking 69
4.1 Problem formulation . . . 70
4.2 Surface patch model . . . 71
4.3 Frame-to-frame non-rigid alignment . . . 73
4.3.1 3D matching of surface patch . . . 73
4.3.2 Analysis of matching error . . . 74
4.3.3 Optimisation of 3D patch matching . . . 79
4.3.4 Weighted Laplacian deformation . . . 82
4.4 Sequential tracking . . . 85
4.5 Evaluation . . . 86
4.5.1 Synthetic facial performance . . . 87
4.5.2 Influence of surface texture . . . 89
4.6 Conclusion . . . 91

5 Robust sequential surface tracking 94
5.1 3D matching of surface patch . . . 96
5.2 Analysis of matching error . . . 98
5.3 Optimisation of 3D patch matching . . . 101
5.4 Coarse-to-fine sequential tracking . . . 104
5.5 Evaluation . . . 106

5.5.1 Synthetic facial performance . . . 108
5.5.2 Influence of surface texture . . . 108
5.5.3 Quantitative evaluation using unwrapped surface textures . . . 111
5.5.4 Variants of frame-to-frame alignment . . . 115
5.5.5 Coarse-to-fine processing . . . 120
5.6 Conclusion . . . 122

6 Non-sequential surface tracking 124
6.1 Problem formulation . . . 126
6.2 Dissimilarity measure . . . 128
6.3 Traversal tree . . . 130
6.3.1 Minimum spanning tree . . . 130
6.3.2 Shortest path tree . . . 132
6.3.3 Cluster tree . . . 134
6.4 Non-sequential tracking using a tree . . . 139
6.4.1 Image-oriented frame-to-frame alignment . . . 139
6.4.2 Geometry-oriented frame-to-frame alignment . . . 140
6.5 Multi-path temporal fusion across tree branches . . . 140
6.6 Evaluation . . . 142
6.6.1 Synthetic facial performance . . . 146
6.6.2 Facial performance . . . 149
6.6.3 Cloth deformation . . . 154
6.6.4 Whole-body performance . . . 156
6.6.5 Results of multi-path temporal fusion . . . 159
6.6.6 Relationship between dissimilarity and alignment error . . . 161
6.6.7 Towards optimal traversal tree . . . 164
6.7 Conclusion . . . 167

7 Facial performance capture 169
7.1 System overview . . . 171
7.2 Capture setup . . . 173
7.3 3D reconstruction . . . 176

7.4 Non-sequential surface tracking 179
7.5 Geometric detail capture 180
7.5.1 Normal map correction 181
7.5.2 Normal mapping 184
7.6 Evaluation 186
7.6.1 Comparison to the state-of-the-art 191
7.7 Conclusion 195

8 Conclusions and future work **197**
8.1 Conclusions 197
8.2 Future work 200

A Marker-based facial performance capture **203**
A.1 Overview 203
A.2 Surface tracking 205
A.3 3D reconstruction 205
A.4 Geometric detail capture 206
A.5 Evaluation 207

B Quantitative evaluation using a mirrored sequence **211**

C Traversal trees **214**

D SEW, SPL and CUT measures **217**

E Facial performance capture - additional results **221**

F Computation time **224**

G Datasets **226**

Bibliography **228**

Notation

Abbreviations

BRDF	Bidirectional reflectance distribution function
CCD	charge-coupled device
CRS	cooperative random sampling
FACS	Facial Action Coding System
HD	high definition
IGD	independent gradient descent
LCS	local coordinate system of a patch
LOD	level of detail in hierarchical surface model
MRF	Markov random field
MST	minimum spanning tree
MVS	multi-view stereo
NCC	normalised cross-correlation
PCA	Principal component analysis
PSCL	photometric stereo with colour lights
PSGI	photometric stereo with gradient illumination
PSWL	photometric stereo with white lights
RGB	red, green, blue
RMS	root mean square
SAD	sum of absolute differences
SNR	signal-to-noise ratio
SPT	shortest path tree
SSD	sum of squared differences
UV	2D parametrisation of surface
VFX	visual effects
WCS	world coordinate system

Mathematical symbols

Symbols used in individual chapters are listed and briefly described. Some symbols are redefined for different chapters.

Typesetting

scalar values	lower-case letters in italic type (e.g. j, η_j)
vectors	lower-case letters in bold type (e.g. $\mathbf{l}_j, \mathbf{r}_i$)
matrices	upper-case letters in bold type (e.g. $\mathbf{L}, \mathbf{V}$)
functions	upper-case letters in italic type (e.g. $E_j(\lambda), A(F_k)$)
sets	upper-case letters in italic type (e.g. X_t, C)
total numbers of elements	upper-case letters in italic type (e.g. J, N)
elements from graph theory	upper-case letters in calligraphic type (e.g. $\mathcal{E}, \mathcal{T}$)
sequences	closed in curly brackets with a range (e.g. $\{M_t\}_{t=1}^{T}$)

Chapter 3

j	index used for light
J	number of lights
I_j	grey-scale image acquired under the light j
$\mathbf{l}_j$	light direction vector
g_j	grey-scale pixel intensity under the light j
$\mathbf{g}$	vector of intensities g_j
$\mathbf{n}$	unit surface normal
α	grey-scale albedo of a surface point for PSWL
$\mathbf{L}$	illumination matrix describing light directions
$\mathbf{V}$	interaction matrix describing light-sensor-material interaction
λ	wavelength
$E_j(\lambda)$	spectrum of the light j
$S_r(\lambda)$	spectral sensitivity of the red camera sensor (similarly for the green and blue sensor)
$R(\lambda)$	material reflectance of a surface point
c_r	pixel intensity in the red image channel (similarly for the green and blue channel)
$\mathbf{c}$	RGB colour
I	colour image
$\rho(\lambda)$	wavelength-dependent chromaticity of the surface
a	grey-scale albedo of a surface point for PSCL
$\mathbf{v}_j$	interaction vector of the red, green and blue sensors with the light j given a surface material
v_{rj}	coefficient from $\mathbf{v}_j$ for the red sensor (similarly for the green and blue sensor)
$\bar{I}_j$	calibration image acquired under colour light j
P	binary mask for the face region in the image $\bar{I}_j$
$\mathbf{p}$	pixel in P

Error analysis uses a simulation-reconstruction chain where subscripts S and R denote elements associated with simulation and reconstruction part. The subscript $*$ represents both S and R.

$\mathbf{L}_*$	illumination matrix
$\mathbf{V}_*$	interaction matrix
$\mathbf{n}_*$	unit normal
a_*	grey-scale albedo
$\tilde{\mathbf{n}}_*$	albedo-scaled normal ($\tilde{\mathbf{n}}_* = a_* \mathbf{n}_*$)
$\mathbf{d}$	difference vector ($\mathbf{d} = \tilde{\mathbf{n}}_R - \tilde{\mathbf{n}}_S$)
$\mathbf{d}_j$	difference vector for the light j ($\mathbf{d} = \mathbf{d}_1 + ... + \mathbf{d}_J$)
$\mathbf{l}_{j*}$	light direction vector from $\mathbf{L}_*$
$\mathbf{v}_{j*}$	interaction vector from $\mathbf{V}_*$
$\mathbf{m}_3$	normalised vector $\mathbf{l}_{1R} \times \mathbf{l}_{2R}$ (similarly for other $\mathbf{m}_j$)
γ_j	angle between $\mathbf{m}_j$ and $\mathbf{l}_{jR}$
β_j	angle between $\tilde{\mathbf{n}}_S$ and $(\mathbf{l}_{jS}^T - \mathbf{l}_{jR}^T)$
$\mathbf{u}_3$	normalised vector $\mathbf{v}_{1R} \times \mathbf{v}_{2R}$ (similarly for other $\mathbf{u}_j$)
δ_j	angle between $\mathbf{u}_j$ and $\mathbf{v}_{jR}$
η_j	angle between $\tilde{\mathbf{n}}_S$ and $\mathbf{l}_j$
$\mathbf{\Delta}$	noise vector from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$
σ	standard deviation of noise
θ_j	slant of the light j
ϕ_j	tilt of the light j
$\triangle\theta$	discrepancy in slant
$\triangle\phi$	discrepancy in tilt

Chapter 4

i	index used for vertices/patches
c	index used for camera/view
t	frame
r	reference frame
T	number of frames
O_t	observations at the frame t
C	number of cameras/views
I_t^c	image at the frame t for the camera c

M_t temporally aligned mesh at the frame t
X_t set of vertex positions of the mesh M_t
Γ set of edges of the mesh M_t
N number of vertices in M_t
$\mathbf{v}_i$ position of the vertex i
V_i set of adjacent vertices around the vertex i
$\mathbf{X}_L, \mathbf{Y}_L, \mathbf{Z}_L$ axes of LCS
$\mathbf{X}_W, \mathbf{Y}_W, \mathbf{Z}_W$ axes of WCS
$\mathbf{p}_i$ position of patch LCS with respect to WCS
$\mathbf{r}_i$ rotation of patch LCS with respect to WCS
$\mathbf{T}_i$ transformation matrix between LCS and WCS
$\hat{\mathbf{p}}_i$ local translation in LCS
$\hat{\mathbf{r}}_i$ local rotation in LCS
$\hat{\mathbf{T}}_i$ transformation matrix for local modification in LCS (formed from $\hat{\mathbf{p}}_i$, $\hat{\mathbf{r}}_i$)
G_i vector of 3D sample points of the patch i
Q_i visibility set of the patch i
B_i^c texture of the patch i from the camera c
d_o 3D distance between rings of sample points in G_i
o index of sample ring
N_o number of sample rings
$E_i()$ patch error function for multi-view alignment
$E_i^S()$ patch error function for multi-view stereo
e_i patch matching error
τ_e threshold for matching error e_i in motion estimate initialisation
V_i^τ subset of V_i for motion estimate initialisation
a_j weight for adjacent motion estimate
$S(I_t^c, \mathbf{p})$ operation projecting a 3D point $\mathbf{p}$ into the image I_t^c and sampling a colour value
$\mathbf{d}_i'$ displacement vector of the patch i (motion estimate)
$\mathbf{d}_i$ displacement vector of the vertex i (final motion)
$\mathbf{L}$ matrix of Laplacian operator
$\tilde{\mathbf{L}}$ matrix of mixed bi-Laplacian and Laplacian operator
μ mixing coefficient for Laplacian operator
$\mathbf{W}$ matrix of constraint weights
s smoothness coefficient
w_i weight of patch displacement
δ_e half-width of linear ramp mapping between e_i and w_i
ξ_e centre of linear ramp mapping between e_i and w_i
M_t^{GT} mesh at the frame t from temporally aligned ground-truth

Chapter 5

M_t^g temporally unaligned mesh at the frame t
X_t^g set of vertex positions of the mesh M_t^g
Γ_t^g set of edges of the mesh M_t^g
$E_i^g()$ patch error function for multi-view alignment and fitting to the mesh M_t^g
σ_g half-width of Tukey bi-weight error norm
$\rho(x, \sigma_g)$ Tukey bi-weight error norm with a variable x
w_g weight of geometry fitting term in E_i^g
H number of iterations in CRS
α ratio of range decrease for random sampling
a integer exponent of α
U number of samples per range size in random sampling

$\mathbf{u}$	random vector from a cube $(-1,1) \times (-1,1) \times (-1,1)$
q_{max}	maximal range for random sampling
q_{min}	minimal range for random sampling
q_{lim}	half size of bounding box limiting possible motion estimates
l	index used for LOD
L	number of LOD in hierarchical surface model
M_t^l	mesh on LOD l at the frame t
G_i^l	sample grid of the patch on the LOD l for the vertex i
V_i^l	set of adjacent vertices around the vertex i on the LOD l
B_i^{cl}	texture of the patch on the LOD l associated with the vertex i from the camera c
ψ_o	scaling factor for patch size N_o across LODs
ψ_s	scaling factor for smoothness coefficient s across LODs

Chapter 6

i, j, u	indices used for frames (respective graph nodes)
r	root frame
k, l	indices used for frame clusters (respective graph nodes)
v	index used for nodes
d	dissimilarity
$\mathbf{D}$	dissimilarity matrix
n	tree/graph node
n_r	root node
$\mathcal{T}$	directed traversal tree among frames
$\mathcal{N}$	set of nodes of the tree $\mathcal{T}$
$\mathcal{E}$	set of edges of the tree $\mathcal{T}$
$\mathcal{G}$	fully-connected undirected graph among frames
$\mathcal{D}$	set of edges of the graph $\mathcal{G}$
$\mathcal{T}'$	undirected spanning tree among frames
$\mathcal{E}'$	set of edges of the tree $\mathcal{T}'$
$\mathcal{T}'_{MST}$	undirected minimum spanning tree among frames
$\mathcal{T}_{MST}$	directed minimum spanning tree among frames
$\mathcal{T}_{SPT}$	directed shortest path tree among frames
F_k	frame cluster
t_k	central frame of the cluster F_k
Δt_k	half-size of the cluster F_k
$A(F_k)$	intra-cluster inconsistency for the cluster F_k
β	granularity parameter for frame clustering
U_β	clustering of frames for given β
K	number of frame clusters
$\mathbf{D}_F$	cluster dissimilarity matrix
$\mathcal{G}_F$	fully-connected undirected graph among clusters
$\mathcal{N}_F$	set of nodes of the graph $\mathcal{G}_F$
$\mathcal{D}_F$	set of edges of the graph $\mathcal{G}_F$
$\mathcal{T}'_F$	undirected minimum spanning tree among frame clusters
$\mathcal{E}'_F$	set of edges of the tree $\mathcal{T}'_F$
$\mathcal{T}'_\beta$	undirected cluster tree among frames given β
$\mathcal{E}'$	set of edges of the tree $\mathcal{T}'_\beta$
$\mathcal{T}_\beta$	directed cluster tree among frames given β
m	length of branch extension across cuts
$\tilde{\mathcal{T}}$	directed tree extended from $\mathcal{T}$ across cuts
$\tilde{\mathcal{N}}$	set of nodes of the tree $\tilde{\mathcal{T}}$
$\tilde{\mathcal{E}}$	set of edges of the tree $\tilde{\mathcal{T}}$
$\tilde{n}_v$	node of the tree $\tilde{\mathcal{T}}$
η_v	blending weight for a tracking solution at the node $\tilde{n}_v$
d_I	dissimilarity for image-oriented frame-to-frame alignment
d_G	dissimilarity for geometry-oriented frame-to-frame alignment

$\bar{e}_I$	average patch matching error for image-oriented frame-to-frame alignment
$\bar{e}_G$	average patch matching error for geometry-oriented frame-to-frame alignment

Chapter 7

$\hat{I}_t^r$	rectified reference image
$\hat{I}_t^m$	rectified matching image
G_r	grid of image points in $\hat{I}_t^r$
G_m	grid of image points in $\hat{I}_t^m$
s_G	size of grids G_r, G_m
P	binary mask for the face region in the reference image $\hat{I}_t^r$
$\mathbf{p}, \mathbf{q}$	pixel in P
V	set defining a 4-point neighbourhood over pixels in P
λ	smoothness coefficient
d	disparity value (horizontal)
$D()$	disparity map for the reference image $\hat{I}_t^r$
$E()$	energy function for the disparity map D

Chapter 1

Introduction

Capture and analysis of human faces is an area of great interest because it has applications in a number of fields. The face is used as a biometric modality for person identification or verification. 3D scans of a patient's face serve as an important aid for surgery planning or prosthesis design. Automatic analysis of facial movement helps to diagnose various medical disorders. Digital avatars controlled by user's expressions are demanded in online communication and telepresence. Game industry, television and film production strive for believable digital characters.

There has been a huge effort in the film industry to create *digital doubles* of real actors. Digital doubles would allow seamless integration of real-world content coming from traditional film shooting with virtual worlds created by computer graphics. The digital double can be placed into environments or situations which would not be possible for an actor. This gives tremendous creative freedom to the director who can influence every aspect of the performance recorded. Moreover, the digital double can be used to produce new performances without the presence of the actor. Development of this technology has a big impact on areas of post-production, facial modelling, texturing and animation. Progress over the last decade has been driven by a number of films such as Final Fantasy: The Spirits Within (2001), The Matrix series (1999 - 2003), The Lord of the Rings series (2001-2003), The Polar Express (2004), Beowulf (2007), The Curious Case of Benjamin Button (2008) and Avatar (2009).

1.1 Motivation

Creating realistic digital double of the face remains a major challenge for computer graphics and animation because of its sheer complexity. The face contains intricate details such as pores, blemishes and fine wrinkles. The skin interacts with incident illumination in a complex way which defines the appearance. Movements of the face are driven by a complicated system of bones, muscles, connective tissues and skin which results in highly non-rigid deformations of the surface. Features such as eyes, hair, teeth or tongue have very different properties compared to the rest of the face.

Creation of a digital model of the face is also complicated by human sensitivity to faces. People observe and analyse faces from an early age because they are the key component of non-verbal communication. Hence, any imperfection or unusual detail in shape, appearance or motion of the face is noticed and can even cause adverse emotions. This phenomenon have been first documented in robotics and is described as the *Uncanny valley*. Mori [75] observed that robots very dissimilar from humans do not trigger any emotional reaction. As their human likeness increases, people find them more familiar and respond positively. But beyond a certain level of robot realism the reaction becomes negative, because there is something uncanny about it which indicates a potential danger. When the robot starts to be completely like a real person, the familiarity sharply increases and people's emotions are positive. This can be depicted as a curve with the Uncanny valley in Figure 1.1. The same observations has been made for virtual characters in computer animation.

Automatic synthesis of believable facial performance for film and television production is impossible with the current technology. Physical simulation of an anatomically correct model of the face is intractable on the level of accuracy required. Thus, digital faces are modelled and animated manually which is a laborious task requiring highly skilled artists. However, it is extremely difficult to avoid the Uncanny valley even for an experienced artist if the digital character is created from scratch. It is common practice to base the character on a real actor and their performance to improve the realism. The artist effectively creates a digital double of the actor.

Actor's performances are recorded and videos can be used as a reference for the artist.

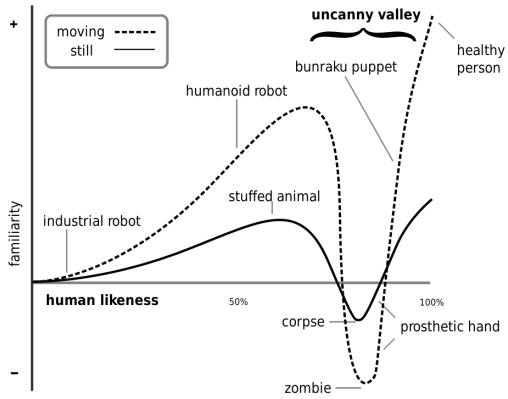


Figure 1.1: Uncanny valley in robotics [75]. Note that movement of the robot emphasises the valley as it is an important cue for judging human likeness.

A better option is to obtain a *4D model of facial performance* representing changes in the 3D shape and appearance of the face over time. The term '4D model' refers to a sequence of reconstructed 3D models with temporal consistency such that points on the models are in correspondence with the same points on the face at each captured frame. Such model is provided by *3D capture systems for facial performances* which can significantly reduce the manual work of the artist required to create the digital double. The 3D models of the actor can speed up or completely replace the modelling phase. Motion of the models during the performance can provide correct facial dynamics for animation. The captured performance can be directly replayed with only manual modification of environmental conditions. Although, this approach improves the quality of digital characters, they still fall into the trap of the Uncanny valley (the film Beowulf being a typical example). Promising results have been demonstrated by The Digital Emily Project [2] or the film The Curious Case of Benjamin Button both of which featured some of the most realistic facial performances by digital doubles so far. But there has not yet been a digital character which would cross the Uncanny valley and significant amounts of manual interaction are still required from artists.

To achieve realistic digital doubles, it is necessary to improve 4D spatio-temporal capturing of facial performance to aid the artist. Thus, facial performance capture needs to advance in two key aspects - *realism* and *temporal consistency* of the resulting 4D performance model.

Realism of the 4D performance model is instrumental for crossing the Uncanny valley. The shape of the face should contain fine geometric details such as skin wrinkles and pores. The appearance of the face should describe the complex interaction of light with skin in terms of diffuse/specular reflectance, translucency and sub-surface scattering. All of this data needs to be captured with adequate temporal sampling to preserve subtle dynamics of the face such as skin wrinkling, pore deformation or changes in skin colouring. This allows high-quality replay of the performance under different conditions defined in post-production. The actor can be relighted according to a virtual environment or can be observed from a novel viewpoint.

Temporal consistency of the 4D performance model means that the facial 3D mesh has a fixed topology and its vertices correspond to the same surface points over time. This introduces a common structure to the data captured in all frames, and therefore allows easy modification of the performance. The temporally consistent model enables space-time editing where a change in one frame can be directly propagated to other frames. This applies to the facial geometry and also appearance associated with it as various texture layers. Without temporal consistency all frames affected by an edit would have to be laboriously modified one by one. Retiming of the performance is another use case for the temporal correspondence across the data which allows interpolation of the facial model for new time instances.

Temporal consistency of the captured data is also important for producing new content. Facial animation rigs are commonly based on blend shapes [84] which require a number of example expressions in full correspondence. This allows weighted blending between them, and therefore creation of new expressions. Animating the rig to generate a new facial performance with the correct motion and timing is difficult and time-consuming even for a highly skilled artist. Therefore, the animation is often driven by a real performance [118] to simplify the process. The temporally consistent representation implicitly provides natural motion of the face over time which is transformed to animation curves for the rig. The curves can be also retargeted to another actor or even a non-human character.

1.2 Objective

The objective of this thesis is capturing high-detail 4D models of facial performances for film production. The model is obtained from multi-view videos of an actor's face recorded in a controlled studio environment. Only the geometry of the face is captured over time because it is more difficult to create realistic facial shapes in motion than appearance. The geometric 4D model of performance should contain all details observed in the acquired videos and have precise temporal consistency.

There is a number of challenges involved in building the 3D capture system for facial performance:

- **Complexity of capture setup** - Existing capture systems often consist of complex equipment such as high-speed cameras, structured light scanners, active illumination elements, etc. All of these components need to be synchronised to provide consistent observations. A practical capture rig should be constructed from standard cameras and lights which does not require complicated integration.
- **Temporal sampling** - Full reconstruction of facial shape often requires multiple observations such as images under various illuminations. Individual observations need to be acquired at high speed to ensure temporal sampling of the whole set at standard frame-rate $25Hz$. A capture system should acquire all necessary data at one time instant, so that a 3D model is available for every frame.
- **Actor's comfort** - An actor typically has to perform under restricted conditions such as a small capture volume or strong fast-switching illumination. Markers or pattern are applied on their face to emphasise motion. Reduction of such constraints would help actors to deliver a more natural performances.
- **Fine skin detail** - Many current methods reconstruct the medium-scale shape of the face with only large skin folds and wrinkles. However, realistic models of the performance should contain pore-level skin detail and fine wrinkles with all subtle deformations over time.
- **Drift** - Temporal consistency of 4D performance models is usually obtained by tracking a 3D facial model in the captured videos. The tracking accumulates errors due to fast non-rigid movements of the face and weak skin texture varying

over time. This results in drift of the 3D model on the actual surface of the face. The proposed system should address drift problems without markers or patterns painted on the face to aid the tracking.

- **Data size** - The level of detail required from a 4D performance model inevitably leads to large amounts of data. A compact representation would be beneficial for efficient storage and fast manipulation.
- **Eyes, teeth, hair** - The face includes features such as the eyes, teeth or hair which have very different properties than the skin. Their accurate capture is a long-standing problem because of the complex geometry, reflectance and motion. However, this problem is not addressed in this work which focuses on skin areas of the face.

1.3 Thesis outline

This thesis is structured as follows:

- **Chapter 2 - Related work**

  A literature review is presented for facial performance capture and specific areas related to the approaches described in this thesis.
- **Chapter 3 - Geometric detail capture**

  Photometric stereo with colour lights is investigated for capturing skin detail. A formulation of the photometric stereo is derived for simultaneous colour illumination. Photometric calibration of a capture setup is explained. Theoretical error analysis of the photometric stereo is supported by simulations on synthetic data. The quality of estimated normal maps is evaluated on real facial performance.
- **Chapter 4 - Baseline surface tracking**

  A baseline method for dense surface tracking is developed based on the work of Furukawa et al. [36, 37]. A patch-based model of the surface is used for tracking a template mesh on multi-view image sequences. 3D matching of a surface patch between frames is formulated and the matching error is empirically analysed. Raw motion estimates from the patches constrain a Laplacian deformation of the template between frames. The method is evaluated for 4D performance capture

with varying amounts of painted texture on the face. This demonstrates accurate temporal alignment with a random pattern but fails for faces without make-up.

- **Chapter 5 - Robust surface tracking**

  A robust surface tracking method is proposed to improve over the baseline method in Chapter 4 on weakly textured skin without make-up. The reformulated objective function for 3D patch matching includes fitting to unaligned per-frame 3D reconstructions. Cooperative optimisation of patches across the surface improves accuracy of motion estimates for Laplacian deformation. Evaluation of the method analyses the influence of varying strength of surface texture. Results for faces without markers or patterns show significantly less drift than for the baseline technique.

- **Chapter 6 - Non-sequential surface tracking**

  A framework for non-sequential surface tracking is introduced which processes an input sequence according to a tree structure across the frames. The traversal tree is derived from the dissimilarity between frames. Several types of trees - minimum spanning tree, shortest path tree and novel cluster tree are described and compared to sequential traversal in terms of the quality of temporal alignment. The framework is generalised to any frame-to-frame alignment method with an associated dissimilarity measure and two different combinations are assessed. Comprehensive evaluation shows temporally consistent mesh sequences with very little drift for facial performances. Versatility of the approach is demonstrated for other non-rigid surfaces such as cloth and whole body.

- **Chapter 7 - Facial performance capture**

  The processing pipeline of the proposed capture system is described and the methods from the previous chapters are tied together. Capture setup for acquisition of multi-view image sequences and subsequent per-frame stereo reconstruction are explained. Non-sequential surface tracking using the cluster tree approach is combined with the photometric stereo with colour lights. Artefacts in the photometric normals are corrected using the temporally aligned mesh sequence. The resulting geometric models of facial performances offer pore-level detail with high-quality temporal consistency which is comparable to the state-of-the-art.

- **Chapter 8 - Conclusions and future work**

  This chapter draws the main conclusions and suggests directions for future work.

- **Appendix A - Marker-based facial performance capture**

  An early marker-based version of facial performance capture system is presented. The approach combines per-frame stereo reconstruction with photometric stereo with colour lights. The resulting 3D models have coarse temporal consistency based on motion of markers painted on the face.

Chapters 4, 5, 6 are closely related because they describe evolution of surface tracking framework. Chapter 3 about acquisition of geometric detail is tied together with the surface tracking in Chapter 7 presenting the whole system.

1.4 Contributions

The main contributions of this work are:

- Error analysis of photometric stereo with colour lights. Normal and albedo accuracy are investigated for image noise, calibration errors in light directions and calibration errors in interaction between lights, sensors and the surface.
- Robust patch-based alignment of a template mesh between two frames. The correspondence is given by cooperative 3D matching of textured surface patches to multi-view images and an unregistered geometry. The method works robustly on plain skin without the aid of markers or pattern make-up. Sequential tracking based on this alignment is more reliable in the presence of fast non-rigid motions than previous techniques.
- Non-sequential surface tracking framework. A template mesh is tracked along branches of a traversal tree calculated from a dissimilarity measure between frames. This greatly reduces drift and impact of failure in comparison to the conventional sequential tracking. The modular framework allows use of any dissimilarity measure, frame-to-frame alignment method and algorithm for calculating the traversal tree. It is possible to align together multiple sequences of the same surface. Versatility of the approach is demonstrated on facial performances, whole-body performances and cloth deformation.

- Cluster tree representation taking into account temporal order of frames to limit the number of alignment jumps introduced by non-sequential traversal. This improves over the minimum spanning tree and shortest path tree used previously in whole-body tracking. The cluster tree enables balancing between sequential accumulation of drift and non-sequential jump artefacts. Remaining jumps in the resulting mesh sequence are eliminated with multi-path temporal fusion.
- Marker-based approach combining photometric stereo with colour lights and stereo reconstruction. This work was proposed to enhance a medium-scale shape from stereo with skin details from the photometric stereo at every frame of a facial performance. Markers painted on the face provide coarse temporal consistency of the resulting mesh sequence.
- Markerless approach combining photometric stereo with colour lights and non-sequential surface tracking based on the cluster tree. This is one of the first techniques using non-sequential traversal for alignment of facial performances. This achieves high-quality temporal consistency of the final models with minimal drift in comparison to previous sequential methods.
- Practical capture setup consisting of several HD cameras and three colour lights. This does not require high-speed recording or active illumination as used in many previous methods.
- 3D capture system for facial performance capture with a full pipeline from capturing an actor to rendering 4D model of a performance.

1.5 Publications

This work has resulted in the following publications:

- M. Klaudiny, Ch. Budd, and A. Hilton. Towards optimal non-rigid surface tracking. In *Proceedings of the European Conference on Computer Vision*, pages 743-756, 2012. (Chapter 6)
- C. Budd, P. Huang, M. Klaudiny and A. Hilton. Global non-rigid alignment of surface sequences. In *International Journal of Computer Vision*, pages 1–15, 2012. (Chapter 6)
- M. Klaudiny and A. Hilton. High-detail 3D capture and non-sequential alignment of facial performance. In *Proceedings of the International Conference on 3D Imaging, Modelling, Processing, Visualization and Transmission*, pages 17-24, 2012. (Chapters 3, 6, 7)
- M. Klaudiny, A. Hilton and J. Edge. High-detail 3d capture of facial performance. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission*, 2010. (Appendix A)
- M. Klaudiny and A. Hilton. Cooperative patch-based 3D surface tracking. In *Proceedings of the European Conference on Visual Media Production*, pages 67-76, 2011. (Chapter 5)
- M. Klaudiny and A. Hilton: High-fidelity facial performance capture with non-sequential temporal alignment. In *Proceedings of the International Symposium on Facial Analysis and Animation*, 2012. (Chapters 3, 6, 7)
- C. Budd, J.-Y. Guillemaut, M. Klaudiny and A. Hilton. Reconstruction and tracking within the SCENE project. In *Proceedings of the Networked & Electronic Media Summit*, 2012. (Chapter 6)

Chapter 2

Related work

This chapter presents a survey of work in the area of facial performance capture. It also includes a brief overview of related work in the areas of photometric stereo and dense motion capture which are important for the approaches developed in this research.

2.1 Facial performance capture

This survey of techniques for facial performance capture is organised according to two key challenges pursued by researchers: *realism* and *temporal consistency* of the 4D model of a performance. Solving both challenges is crucial for the creation of a digital copy of a performance for use in film and television production.

2.1.1 Realism of the 4D performance model

Systems for facial performance capture differ in the amount of shape and appearance detail present in the 4D performance model acquired. This section categorises the systems according to underlying methods for shape reconstruction - *stereo*, *structured light*, *shape from shading*, *photometric stereo with colour lights* and *Light Stage*. Some of the systems combine two methods but they are categorised according to the method providing finer geometric information.

Stereo

Stereo reconstruction has been widely used for obtaining 3D models of the face [36, 19, 122]. A capture rig in the minimal configuration needs only two standard cameras. Typically, more cameras together with lighting elements are used to increase facial coverage and reconstruction quality. Facial shape can be reconstructed by matching image patches in stereo camera pairs and fusing the resulting depth maps into a single mesh. Alternatively, the whole surface is computed at once by multi-view stereo (MVS) which jointly matches the image patches across views.

Passive stereo techniques rely on the facial appearance under diffuse white illumination for 3D reconstruction and tracking. Therefore, they do not use any active lighting elements and 3D models of the face can be acquired at camera frame-rate. Furukawa et al. [36, 37] paint random pattern on the face to enhance weak skin texture in $1Mpx$ images. This yields accurate mesh sequence exhibiting skin wrinkling but natural appearance of the face is obscured. An alternative to the pattern is an increase of the camera resolution, such that skin pore structure of skin is clearly visible. Bradley et al. [19] construct a rig with 7 pairs of HD cameras zoomed on overlapping regions of the face. The acquired high-resolution skin detail allows reconstruction of the geometry comparable to the pattern-based method [37]. In addition, the meshes have extremely detailed dynamic texture ($10Mpx$). DI4D capture system from Dimensional Imaging [31] achieves similar fidelity of the facial model with less cameras.

Before the use of high-resolution cameras the weakness of skin texture was overcome by projecting a random pattern on the face to improve stereo matching. But this complicates simultaneous appearance capture. Zhang et al. [122] introduce space-time stereo aided with a projected pattern to obtain accurate depth maps with little temporal noise. Every third frame is recorded with full illumination to allow surface tracking and appearance capture. Small deformations are not captured due to the lower resolution of the tracked template than depth maps (Figure 2.1(b)). The texture sequence also lacks details due to the use of 640×480 camera sensor. Two pairs of grey-scale cameras for shape acquisition and two colour cameras for appearance acquisition operate at $60fps$ and full textured meshes are produced at $20fps$. Commercial system Mova

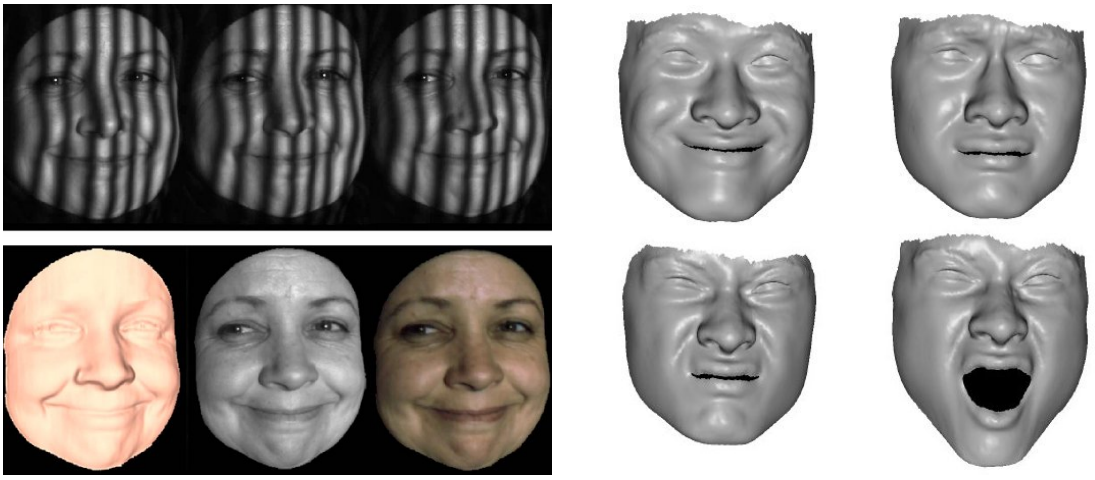
CONTOUR Reality Capture [76] uses a random pattern painted on the face to improve MVS reconstructions at each frame. The pattern is fluorescent and invisible under white light which allows joint capture of the shape and appearance by fast switching between white and ultra-violet light. The geometries are aligned to a medium-resolution mesh sequence textured with dynamic high-resolution colour map. To achieve truly simultaneous capture of the shape and appearance, the pattern can be projected in the infra-red part of light spectrum [121, 1]. 3dMD dynamic system [1] has two pairs of infra-red cameras which provide depth maps for each side of face at $60fps$. They are synchronised with two colour cameras which simultaneously obtain textures.

Structured light

Structured-light systems typically project a series of fringe patterns on the actor [124, 115]. The patterns temporally encode planes across 3D space which combined with rays cast from the observing camera define 3D surface points. This does not allow capture of shape at video frame-rate and simultaneous appearance acquisition. High frame-rate cameras are required to interlace the pattern projections for geometry and white illumination for colour texture. This results in a temporal offset between shape and appearance capture. A capture rig typically contains a light projector, a grey-scale camera recording the projected patterns and a colour camera recording facial appearance.

Zhang et al. [124] proposed the phase-shifting method which cycles only three sinusoidal fringe patterns at $120Hz$. 3D reconstruction is performed in real-time for each pattern cycle independently. Colour texture is obtained using a long exposure on the colour camera over the multiple patterns. The system provides textured meshes with temporal sampling of $40fps$ (Figure 2.1(a)). Wang et al. [112] fits a template mesh to geometries from Zhang's scanner. The final performance model contains large skin folds and wrinkles and has a low-resolution dynamic texture. Walder et al. [111] achieves a similar quality of shape but better appearance due to higher image resolution of cameras in their scanner. Weise et al. [115] present for a system with an additional grey-scale camera which combines phase-shifting with stereo matching to improve depth

discontinuities. Partial scans of the face are produced at $25 fps$. An actor-specific PCA deformable model with fixed appearance is fitted to the scans and images but its shape space does not model any skin wrinkles. Li et al. [64] enhance a warped template mesh with details over time. Shape details such as medium-sized wrinkles are encoded as displacement coefficients along vertex normals. They dynamically change over time but the stable features are gradually aggregated and preserved in the model. Thus, these features are present in the mesh even if a current scan does not contain them.



(a) (b)

Figure 2.1: Structured-light method [124] - fringe images, a shaded and textured model of the face (a). Stereo-based method [122] fitting a template mesh to depth maps (b).

Shape from shading

Shape from shading derives surface normals and reflectance of an object lit by directional light from a single image. This is inherently an under-constrained problem for a single surface point, therefore various global regularisation constraints have been introduced [123]. General regularisation terms such as surface smoothness do not yield correct shape for faces. Thus, statistical model of facial normals is used as a shape prior for the regularisation of normal map [95, 96, 97]. Another option for the prior can be 3D morphable facial model [83]. Local shading constraints from pixel intensities can be modelled according to different reflectance models - Lambertian [95], Torrance-Sparrow[96] or general bidirectional reflectance distribution function (BRDF)

[97]. Facial shape integrated from normals contains larger wrinkles but no pore-level skin structure. All mentioned methods provide static capture from a single viewpoint.

Dark-is-deep concept represents an approximation of shape-from-shading formulation, such that a darker/brighter pixel intensity indicates a valley/bump on the surface [63, 42]. This is valid especially under diffuse white illumination which is problematic for standard shape from shading assuming collimated light. The reason is that a pixel intensity depends more on the amount of light reaching a surface point than on a surface normal. A large intensity change with respect to neighbourhood means that the surface point is largely occluded and is inside a valley. Based on this simplified interaction between illumination and surface, it is possible to hallucinate fine details such as skin wrinkles or pores. Their shape can be be visually pleasing but is not accurate.

Passive capture systems use dark-is-deep concept to enhance the medium-scale meshes with approximate skin detail. This approach is suitable because passive setups typically rely on white diffuse illumination. Bickel et al. [14] build a high-detail model of the neutral face using 3dMD system [1] for base shape and the method by Weyrich et al. [117] for normals and skin reflectance. This model is then deformed throughout the performance using markers painted on the face. Shape of large wrinkles is inferred from shading under diffuse illumination at every frame. They are painted by distinct diffuse colours to simplify their tracking and improve the shape inference (Figure 2.2(a)). The reconstructed wrinkles dynamically enhance the template model but fine skin geometry and appearance remains fixed over time. Borshukov et al. [17] used a similar approach for facial performance capture in The Matrix sequels but without use of any make-up. A high-detailed laser scan of the neutral expression is stripped of fine details which are stored in a static bump map. Medium-sized wrinkles are extracted from shading in the images and added onto the scan as a dynamic displacement map. Five HD cameras capturing the performance provide time-varying high-resolution texture map.

Beeler et al. [10] present a static capture system which acquires images of an actor under white diffuse light by 7 DSLR cameras. MVS reconstructs a base shape of the face which is further refined by the dark-is-deep approach. The approximate mesoscopic layer



(a) (b)

Figure 2.2: Marker-driven technique enhanced by dark-is-deep concept [14] - one input image, a deformed template mesh, a template with added wrinkles and a final textured model (a). High-quality MVS reconstruction augmented with dark-is-deep approach [10] - one input image, a shaded and textured final mesh (b).

displaces vertices of the base mesh to include small wrinkles and pores (Figure 2.2(b)). This system has been extended to dynamic capture using 7 cameras with frame-rate $46fps$ [13]. The output is a high-resolution mesh sequence which contains fine skin deformations.

Photometric stereo with colour lights

Hernandez and colleagues introduced a photometric stereo with colour lights (PSCL) which obtains normal maps of dynamic surfaces [20]. An actor is simultaneously lit by red, green and blue directional light from different angles. A normal map is computed for every frame captured by a single camera without any time-multiplexing of the illumination. An assumption of Lambertian surface with constant albedo requires application of uniform make-up on the face. Integration of normal maps results in detailed but distorted facial shape. This is a consequence of low-frequency bias in normals which is a common issue of photometric stereo methods. Self-shadows due to directional illumination also bias affected normals. A simple correction in [20] can handle pixels with one occluded light assuming constant albedo. Hernandez et al. [49]

offer more robust scheme for the shadow correction which permits varying grey-scale albedo but uniform chromaticity. A normal map is optimised jointly to enforce surface continuity in shadowed regions.

Vogiatzis et al. [110] alleviate restrictions on the reflectance by incorporating Phong model. Chromaticity and specular reflectance parameters are assumed constant but grey-scale albedo can vary across the face. Photometric calibration selects a dominant chromaticity on the face, so that normals are calculated precisely for most of the surface. Integration of normal maps produces visually plausible meshes with fine skin detail even for faces without uniform make-up (Figure 2.3). However, regions with other than the dominant chromaticity are distorted to some extent. Anderson et al. [3] combine PSCL with depth acquisition by Kinect sensor [72]. Available depth maps enable normal calculation for multiple chromaticities because image pixels can be associated with a particular chromaticity based on orientation estimate. The obtained normal maps augment the depth maps with geometric detail at every frame using [77].

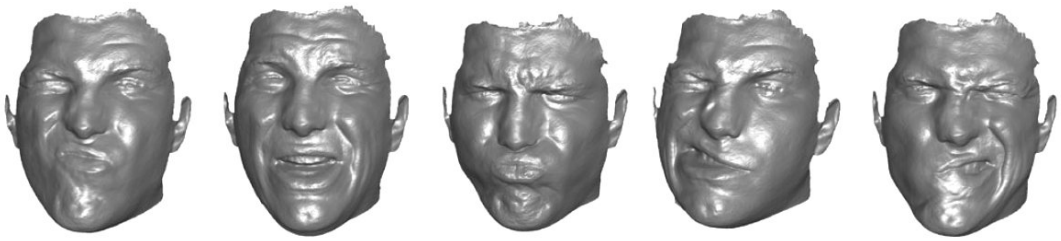


Figure 2.3: Meshes integrated from normal maps by PSCL [110].

Light Stage

Debevec and colleagues introduced the Light Stage technology [27, 116, 69, 39] which captures an actor under many different illumination conditions. High-resolution normal map and reflectance of the face are calculated from changing appearance due to the illumination changes. This is facilitated by complex lighting setup synchronised with typically a single camera which records the face during cycles of illumination patterns at high frame-rate. Effective frame-rate of the resulting facial models depends on the number of illumination patterns and the camera frame-rate.

The first Light Stage approach [27] records an actor in still pose while a single light source moves around on a spherical trajectory for one minute. This samples facial reflectance under 2048 light directions and allows relighting of the actor's pose under different environmental illumination. Hawkins et al. [45] improve the capture process by rotating an arc of lights around the actor in $8s$. Reflectance of the face is scanned under 480 light directions for 60 different facial expressions and visemes. A blend-shape model of the face is constructed together with the sampled reflectance which allows relighting of the actor for any created pose. Actor's performance can be captured by deforming the model according to motion of markers on the face.

Wenger et al. [116] take the Light Stage concept to truly dynamic capture. A spherical light dome with 156 LEDs enables quick changes between complicated illumination patterns (Figure 2.4(a)). The dome is synchronised with high-speed camera recording cycles through different patterns at $2160fps$. A Hadamard illumination basis is used instead of a cycle through individual lights to give improved signal-to-noise ratio (SNR) by capturing the face under greater illumination for each sample image. Actor's motion during capturing the full basis is compensated by optic flow between fully lit tracking frames inserted at $120Hz$. Surface normals together with diffuse albedo and ambient occlusion are computed from the sampled reflectance. A full set of observations is acquired at effective rate of $24fps$ for very short performances which can be realistically relighted afterwards.



(a) (b)

Figure 2.4: The Light Stage capture rig from [2] (a). A static capture with the polarised Light Stage [67] - a high-resolution mesh and renderings with reflectance (b).

The previous methods do not allow relighting with directional spatially-varying illumination because a facial shape is not recovered. Jones et al. [59] address this problem by combining the reflectance sampling [116] with the structured light. A scanner projects 24 horizontal and vertical patterns to reconstruct the mesh. The Light Stage illuminates an actor by 29 basis conditions which are used in photometric stereo to obtain diffuse normals and albedo. The whole system runs at $1500fps$ and outputs high-resolution meshes at $24fps$. Motion correction between the basis images is not applied in contrast to [116], hence speed of facial movements is restricted. A static capture method of Weyrich et al. [117] combines the Light Stage with stereo reconstruction [1]. The face is recorded by 16 cameras during illumination cycles through 150 light directions. This yields a comprehensive model of the face consisting of diffuse and specular normals, diffuse albedo map and specular BRDF map. Parameters of sub-surface scattering in the skin are measured separately by a fibre optic spectrometer. This results in highly realistic renderings of the face.

To reduce a number of illumination conditions, Ma et al. [67] propose a new photometric stereo technique based on spherical gradient illumination (PSGI). Three illumination conditions - X, Y, Z gradients are required for normal calculation and a constant illumination provides colour albedo (Figure 2.5(a)). Because gradient illumination is specularly reflected across the whole face, it is possible to compute dense normal and albedo map only from specular component of the light reflection. Specular normals are more accurate than diffuse ones because the light is reflected from the outer surface of the skin and is not subject of sub-surface scattering. Diffuse and specular component of the reflection are separated using linearly polarised illumination and a polariser switching between parallel and cross orientation in front of the camera. The obtained normal and albedo map are combined with a mesh from pattern-aided stereo to create realistic static model of the face (Figure 2.4(b)).

To extend the combination of pattern-aided stereo and PSGI to dynamic capture [69], polarisation cannot be used due to technical limitation of switching the camera polariser at high speed. Therefore, photometric normals embossing the mesh reconstructed by the stereo [77] are less crisp. The textured meshes in Figure 2.5(b) are acquired at $24fps$ by processing groups of 12 images under different light patterns (a subset in

Figure 2.5(a)). The images are motion-corrected by optic flow between constantly lit frames assuming linear motion. These techniques [67, 69] have been used for building high-detail blend-shape rig in The Digital Emily Project [2]. Realistic digital copy of a real performance is created by driving this rig according to an original video.

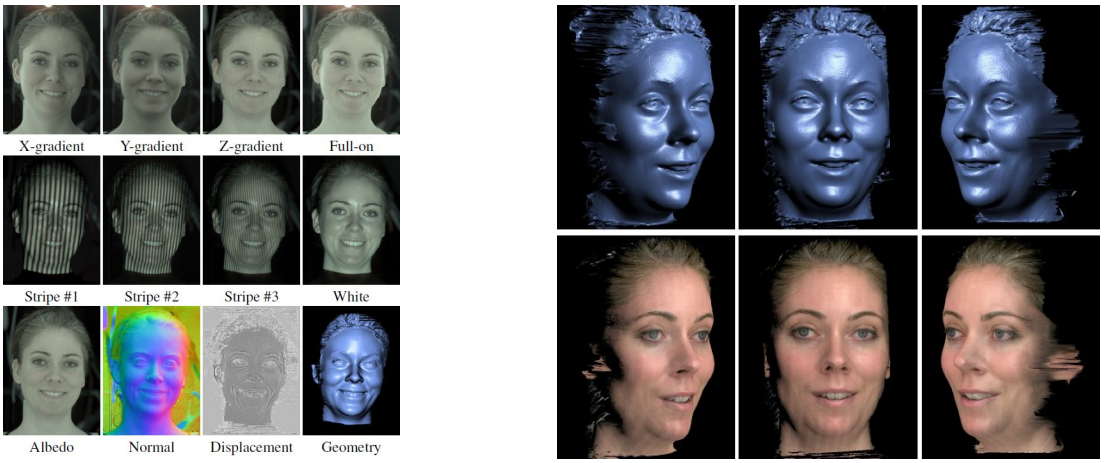


Figure 2.5: A combination of PSGI and structured light by Ma et al. [69] - gradient images, a subset of stripe patterns and different layers of the model (a); a shaded and textured final model (b).

Wilson et al. [119] improves accuracy of normals on darker side of gradient patterns in PSGI by introducing complement gradient patterns with opposite direction. Every tracking frame with constant illumination is flanked by three original X, Y, Z gradients and their three complements. The pairs of complement frames are jointly aligned with the tracking frame using an optic flow algorithm handling gradient shading. This enhances crispness of normal map which is merged with 2.5D mesh computed by stereo at the tracking frame. Replacement of the structured light with the stereo reconstruction reduces a number of images required for a full 3D model in contrast to [69]. Direct temporal alignment of gradient images also enables accurate model interpolation between the tracking frames, so that full facial models are available at every frame captured. Fyffe et al. [39] extend [119] with MVS which improves fine details around eyes and mouth. Also, better coverage of the face is achieved with 5 high-speed cameras ($264fps$). Diffuse and specular reflectance are heuristically separated, so that normals, diffuse and specular albedos can be obtained. Final models of the face are available

at standard $24fps$. Drawbacks of the diffuse/specular separation through polarisation [67] are alleviated in the static capture system by Ghosh et al. [41]. Instead of mechanical rotation of a camera polariser, linear polarisation of the illumination changes between latitude and longitude orientation by alternating two sets of lights. This also lifts a single-view restriction imposed by a fixed polarisation pattern across the Light Stage which is optimised for the specific camera position. Thus, multi-view capture can be combined with the polarisation which improves MVS by utilising diffuse/specular normal maps.

Discussion

The described categories of performance capture systems differ in several important key listed in Table 2.1. They are compared in terms of lighting conditions during capturing, level of geometric detail in the facial 3D model, type of appearance data acquired and the number of observations required for obtaining one instance of the model.

Category	Lighting conditions	Geometry	Appearance	No. of observations
Stereo	diffuse white illumination	medium-scale facial shape	colour texture	1
Structured light	multiple fringe patterns	medium-scale facial shape	colour texture	3-4
Shape from shading	directional/diffuse white illumination	approximate skin structure	BRDF / colour texture	1
PSCL	directional colour illumination	accurate skin structure	grey-scale albedo	1
Light Stage	multiple illumination patterns	accurate skin structure	BRDF	7-156

Table 2.1: Categories of capture systems according to realism of the 4D facial performance model.

Stereo-based approaches usually use a random pattern painted or projected on the face to improve the reconstructed shape. Use of high-resolution cameras in recent methods has eliminated need of the pattern because skin structure provides enough image information for accurate matching. However, the camera resolutions are not large enough to reconstruct pore-level details. A capture rig can be constructed from

standard cameras and lighting equipment. The advantage of the stereo is reconstruction of a 3D model at every frame. Passive white illumination also enables simultaneous acquisition of colour texture of the face.

Inaccuracy of stereo matching on skin texture under low image resolution was circumvented using *structured-light* 3D reconstruction at the expense of active illumination. This requires several observations with different fringe patterns projected on the face at high frame-rate which can be distractive for an actor. Facial colour texture can be captured in additional frame or extracted from the pattern frames. 3D models of the face contain shape details up to larger skin wrinkles which is comparable to the current stereo-based systems.

Standard *shape from shading* has been used for static facial capture under a single directional light source. Statistical model of the face used as a regularisation prior enables reconstruction of facial shape up to large wrinkles and estimation of different parametric BRDFs. For diffuse white illumination the dark-is-deep concept approximates the shape-from-shading approach which is useful for dynamic capture systems with passive illumination. At each frame the estimated normal maps and colour textures enhance medium-scale facial meshes from less accurate 3D reconstruction methods. This provides approximate skin structure which is visually pleasing but less accurate than photometric stereo.

Approaches using *photometric stereo with colour lights* acquire accurate pore-level geometric detail at every frame. However, the obtained normal maps contain low-frequency bias which distorts facial meshes integrated from the normals. PSCL has also a restriction of single or few constant chromaticities across the surface which is not entirely valid for the face. This allows only acquisition of grey-scale albedo maps. The capture setup contains three passive directional colour lights which is considerably simpler than the Light Stage.

Light Stage systems capture accurate facial shape up to fine skin details together with extensive reflectance data. This requires complex capture rig alternating between many illumination conditions which can be uncomfortable for the actor. High-speed cameras record cycles of the conditions to ensure standard frame-rate of full facial models. Use

of the Light Stage for photometric stereo with gradient illumination has reduced the number of illumination patterns to seven and therefore necessary camera frame-rate has decreased. However, optic-flow alignment of images under different illumination patterns is still necessary and can be imprecise under fast facial motion.

2.1.2 Temporal consistency of the 4D performance model

Systems for facial performance capture can be divided into several categories depending on how they achieve temporal consistency of 4D performance model. This section describes *independent per-frame 3D reconstruction*, *3D deformable models*, systems using *markers*, a dense *pattern*, *geometry-based alignment* and *image-based alignment*.

Independent per-frame 3D reconstruction

Static facial reconstruction performed independently at regular time steps is the simplest way of capturing dynamic performance. However, lack of temporal consistency in the data limits use to replay and does not allow spatio-temporal editing. For relighting the actor in the video according to a new virtual environment, it is sufficient to obtain time-varying surface normals and reflectance [116, 67]. To change the viewpoint or cast shadows, the full 3D shape of the face needs to be reconstructed. This can be obtained by stereo matching at every frame of multi-view image sequences [121, 1]. An alternative for reconstruction of the medium-scale geometry is a structured-light approach [124]. Ma et al. [69] and Jones et al. [59] improve the geometric resolution of structured-light reconstruction with normal and reflectance data provided by PSGI. A similar enhancement with the photometric detail can be done for MVS reconstructions [39]. Vogiatzis et al. [110] compute dynamic normal maps using PSCL and integrate them into meshes in real-time.

Deformable models

Temporal consistency over a performance can be achieved using a 3D deformable model of a face which is fitted to the captured data. A single mesh topology of the facial model

and its possible deformations are designed before the performance alignment. Model deformations are controlled by a set of parameters which is significantly smaller than a number of mesh vertices. Thus, there is far less variables to optimise during the model fitting to data constraints coming from images and/or geometry. Global optimisation across the whole model helps to overcome missing strong constraints for parts of the face (e.g. smooth skin regions).

Widely used deformable models for the face are based on a dataset of example facial shapes which represents a space of possible deformations. These examples typically include various facial expressions, visemes and eye poses. This is often inspired by Facial Action Coding System (FACS) [35] which describes all possible states of the face. The dataset can include faces of multiple people to create a person-independent model. All example meshes are usually textured, so facial appearance can change together with shape. A new instance of the face is created as a linear combination of the examples comprising the model. Blending of the textured meshes is possible only if they are in full correspondence. Mesh registration across different facial expressions obtained by static 3D capture is complicated and often semi-automatic process.

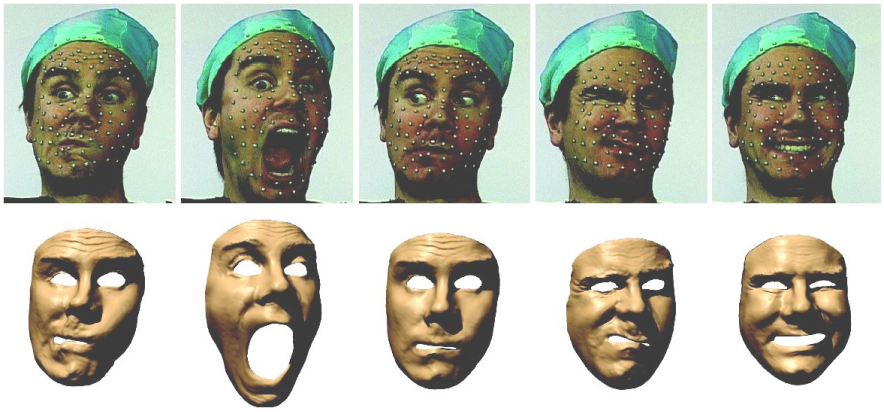


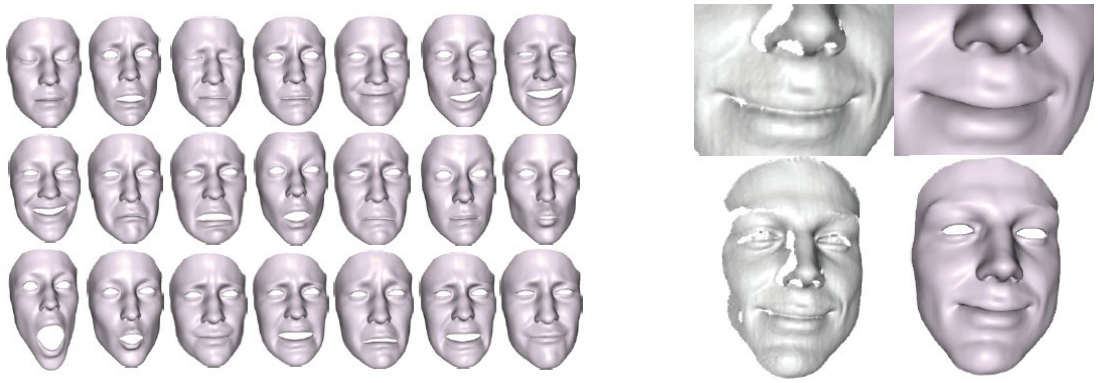
Figure 2.6: Blend-shape model driven by markers [51].

The first type of linear statistical deformable model is a blend-shape model used in facial animation [84]. Control parameters are blending weights used for interpolation between example shapes. Because they are directly associated with them, they have semantic meaning interpretable by an animator. The model is typically person-specific due to nature of the application. The blending weights can be estimated for all frames

of a real performance to aid the animation process or just track the actor's face. This is often based on motion of markers applied on the face [45, 51]. Huang et al. [51] analyse the marker motion of a recorded performance to find the most compact set of example expressions representing the data (Figure 2.6). The model is built from high-detail laser scans which enable capturing medium-sized wrinkles. Although the dataset is optimised for a particular performance, details are not completely correct for some blended poses. Other methods exploit full image information instead of sparse marker locations. Pighin et al. [85] optimise the weights so that rendering of the interpolated textured mesh resembles a target image. To increase flexibility of the blend-shapes, the face is also split into regions which are deformed separately. Alexander et al. [2] has achieved impressive results with semi-automatic image-based method from Image Metrics [57] which drives the blend-shape model by a single-view video.

The second type of linear model is the 3D morphable facial model by Blanz and Vetter [15] which is derived from 2D Active appearance model [25]. Similarly to the blend-shapes, it is built from a dataset of faces which vary in expressions and also in a person's identity. Control parameters are derived by Principal component analysis (PCA) on the input dataset. This allows a reduction of parameters by selecting a number of the most significant principal components. Although, PCA coefficients influence the face independently from each other, they lack semantic meaning. This can be alleviated by mapping them to semantic facial attributes but this involves manual labelling of the examples. Fitting the morphable model to a 3D laser scan or images from several viewpoints in [15] showed a potential for the facial performance capture. This is fully demonstrated by recent work of Weise et al. [115] where a person-specific model is tracked on-line on the raw geometries from a real-time structured-light scanner. An extension [114] using the Kinect sensor [72] tracks the model on depth maps and images simultaneously. Optic flow constraints from a single-view video stabilise the fitting in the presence of high noise levels in the captured geometries.

The third type of linear model is based on multi-linear algebra [108]. It is created from a set of 3D scans [1] of various people performing different expressions and visemes. The model has three separate modes for identity, expression and viseme which provide independent groups of control parameters not influencing each other. During facial



(a) (b)

Figure 2.7: A subset of facial expressions used for building a 3D morphable model [115] (a). Example of fitting the model to geometry from a structured-light scanner (b).

tracking only the expression and viseme parameters are estimated according to optic flow constraints from 1000 selected mesh vertices. The identity mode of the multi-linear model can be used for transfer of the performance to another actor.

The 3D deformable model can be also based on physical deformation operators instead of shape interpolation in an example space. Control parameters then describe various types of mesh deformations such as scaling, stretching, bending etc. DeCarlo et al. [28] proposed a multi-part facial model where different sets of operations are defined by a user over individual parts or groups of the parts. The deformation parameters are directly incorporated into a model-based optic flow which estimates motion of the face. Huang et al. [55] uses a multi-resolution model for the whole face where a coarse mesh is deformed by simple operations to capture large movements. This is refined on dense resolution by Free Form Deformations which are better suited for local non-rigid motions. The model is tracked according to point clouds from a structured-light scanner.

Markers or pattern

Free-form tracking of the face is desirable because deformations are not restricted by any prior model. However, finding reliable correspondences for all vertices of a template mesh is difficult and the mesh usually starts to drift on the face after short period

of time. The reasons are fast non-rigid deformations and relatively uniform skin appearance in standard image resolution. A pragmatic solution to this problem has been the application of markers pioneered by Williams [118] which is still widely used in industry.

Up to several hundred passive markers are glued or painted on the face. They can be observed by several cameras [44, 70, 14], a single camera combined with side-view mirrors [66] or head-mounted camera rig (VICON Cara [107]). Because of their distinct appearance it is much easier to find correct correspondence across views and over time in comparison to the skin. Correspondence algorithms have to be robust against occlusions and mismatches due to fast motion. This is addressed by adaptive Kalman filters [66] or graph matching [44]. 3D trajectories of the markers drive deformation of a template mesh over time (typically a laser scan of the face). The resulting temporally aligned meshes do not contain small surface deformations because of the limited number of markers. This is partially alleviated by Bickel et al. [14] who track large wrinkles and synthesise their shape in the final mesh sequence in post-processing (Figure 2.2(a)). Because the markers occlude the natural appearance of the face, the majority of methods capture the texture and reflectance data beforehand [66, 70, 14]. This results in static appearance over time which is not visually pleasing. Guenter et al. [44] obtain time-varying textures by erasing the markers and filling holes with synthesised skin.



Figure 2.8: Temporally consistent mesh from a pattern-based method [37] - a sample image, shaded mesh, motion field and a textured mesh.

To achieve better sampling of facial motion, a dense random pattern can be painted on an actor instead of markers. Work by Furukawa and Ponce [36, 37] tracks a dense mesh reconstructed by MVS in the initial frame. The mesh is deformed directly according to

multi-view image sequences without prior computation of unregistered meshes or optic flow fields. The results show accurate temporal alignment of medium-sized wrinkles. The Mova CONTOUR Reality Capture system [76] uses fluorescent pattern visible under ultra-violet light to improve optic flows computed across multiple cameras. The multi-view flows together with MVS meshes accurately drive deformation of a user-designed mesh.

Geometry-based alignment

Instead of enhancing skin texture by markers or pattern, some approaches achieve temporal consistency by focusing on the shape of the face. Input data are typically unregistered point clouds or meshes coming from structured-light scanners. The geometries often carry colour texture which can provide sparse constraints from strong appearance features [112, 87]. The raw meshes in successive frames can be cross-parametrised by mapping into the 2D domain and aligned there using the appearance constraints. Wang et al. [112] unwrap the whole meshes onto a 2D disk using harmonic maps. Popa et al. [87] segment the surfaces into regions which are unwrapped by local low-stretch maps. The region-based approach allows handling topology changes and missing data. Drift is restricted by hierarchical merging of sub-sequences which are aligned independently.

Li et al. [64] warp a template mesh in 3D space without any appearance constraints. The warping is controlled by a deformation graph associated with the template which is refined over time to accommodate new deformations observed in per-frame scans. In contrast, motion of template vertices can be optimised directly based on the distance from scans, acceleration, surface rigidity and colour variance as in [111].

Image-based alignment

Many techniques infer temporal correspondence from facial appearance rather than shape because there is intuitively more information variance. Especially with increasing resolution of cameras, the captured skin texture starts to contain fine details such as pores and blemishes which create distinct patterns. However, it is still challenging

to match skin patches over time because their appearance changes dramatically with surface deformations. This commonly leads to drift during temporal alignment.

The input is image sequences from multiple cameras which are first processed separately to obtain frame-to-frame 2D optic flows (widely used Brox's method [82]). A template mesh is fitted to the first frame and is sequentially propagated between frames according to the multi-view flows [17, 93]. Sibbing et al. [93] compute sparse optic flow using 2D mesh where displacements are interpolated within mesh triangles. Surfel anchors attached to the template are moved to the next frame according to the flows but their position and orientation is refined afterwards according to MVS objective. The technique used for the Matrix sequels [17] includes a semi-automatic process for correcting pre-computed optic flows to meet high quality of temporal consistency required.

The majority of methods also reconstruct facial shape using MVS at every frame. The 3D reconstructions do not provide motion estimates as in the geometry-based approaches but they rather constrain deformation of the template to the correct shape. Vertices of the template mesh are moved according to flow fields in individual views and conform to unregistered geometries [31, 122, 19, 13]. Zhang et al. [122] aid stereo with a projected pattern but every third frame is recorded without it to allow computation of optic flow. Deformation of the template is defined as a global optimisation over all vertices. Data terms force the vertices to follow flow fields and stay close to depth maps. A regularisation term penalises different motion in adjacent vertices.

Passive performance capture techniques [31, 19, 13] rely on a high image resolution to acquire enough skin detail for reliable surface tracking (Figure 2.9). Frame-to-frame alignment of the template mesh in [19, 13] differs from Zhang et al. [122]. Each vertex is projected into individual views to obtain 2D positions in the next frame from flow fields. The new 2D positions are back-projected onto a raw mesh in the next frame and fused into a new vertex 3D position. After displacing all vertices separately the whole mesh is regularised by Laplacian deformation to filter outliers. Quality of the alignment and shape of small features can be improved by cancelling ambient occlusion in concave regions (e.g. valleys in between wrinkles). Beeler et al. [12] remove estimated ambient occlusion from input images. The recalculated optic flow is more accurate because skin

appearance change is smaller between frames. The shape is refined according to the difference between the current frame and a neutral expression warped by optic flow.

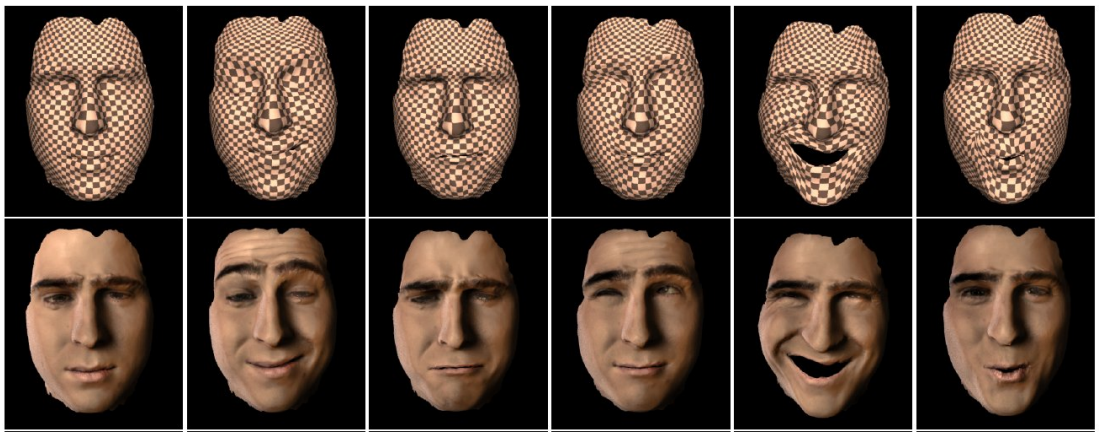


Figure 2.9: A temporally consistent mesh sequence from [19] - a fixed UV texture (a) and time-varying real appearance (b).

Systems capturing the geometric structure of the skin exploit this for temporal alignment because of richer detail than skin appearance [68, 119]. The combination of original images with normal/displacement maps improves accuracy of optic flow. Ma et al. [68] unwrap meshes from a structured-light scanner in a common texture space using several markers on the face. Each mesh is textured with a fine displacement map from PSGI which is used for dense mesh alignment by optic flow in the texture space. Wilson et al. [119] combines PSGI with stereo reconstruction using one HD camera pair. Optic flow is calculated between full-lit tracking frames in one view leveraging also normal maps which are computed for them from adjacent gradient-lit frames. The full-lit images, normal maps and 2.5D meshes are interpolated between the tracking frames to provide complete models for all captured frames.

Despite the high image resolution and the augmentation with skin geometry, motion estimation still contains some amount of error. Sequential concatenation of frame-to-frame estimates inevitably leads to the accumulation of errors (e.g. significant drift after ~ 200 frames in [119]). Bradley et al. [19] apply drift correction on the resulting temporally consistent mesh sequence. Their assumption is that textures projected on the meshes in every frame should be stable in texture space if the temporal alignment has been accurate. Additional optic flow computed in the texture space measures

shifts with respect to the initial frame and enables correction of the drift in 3D space. However, there is still some amount of inaccuracy around lips which undergo the most complex deformations (Figure 2.9). Non-sequential tracking of the face is proposed by Beeler et al. [13] which relies on the frequent occurrence of the neutral expression in any performance. The neutral expression is detected throughout the sequence and so-called anchor frames are created. The template mesh is first tracked between the initial frame and the anchor frames which contain similar neutral expression. The tracking then continues sequentially between the anchor frames. This significantly limits the possibility of drift due to shorter chains of frame-to-frame alignments towards each frame. Impressive results demonstrate temporal consistency up to skin pore level (Figure 2.10). The non-sequential approach is adopted by the commercial DI4D capture system developed by Dimensional Imaging [31]. Anchor frames are selected by a user which allows flexibility and control in terms of traversal through the data.

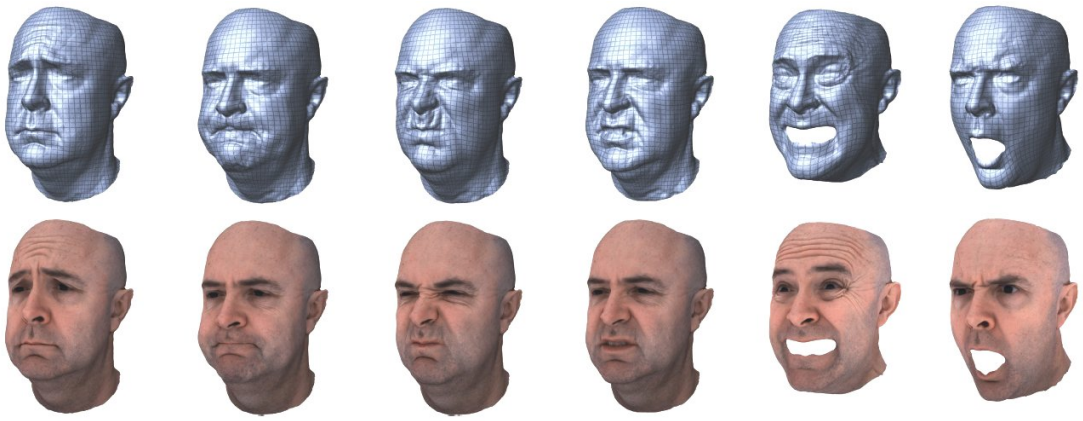


Figure 2.10: A temporally consistent mesh sequence from [13] - a fixed UV texture (a) and time-varying real appearance (b).

Discussion

The described categories of capture systems differ in terms of temporal alignment used. Temporal consistency of the 4D performance model has varying accuracy depending on the amount of drift and resolution of motion estimation. *Independent per-frame 3D reconstruction* of the face is the simplest way of capturing dynamic performance but there is no temporal consistency in the captured data.

Fitting a *3D deformable model* to the input data at every frame implicitly yields temporal consistency across the frames. The deformable model of the face needs to be created before performance tracking but it is deformed according to a relatively small number of parameters compared to the number of mesh vertices. Example-based deformable models require building a dataset of example facial shapes which need to be fully registered. Another limitation is capturing facial expressions which are quite different from the example dataset. Also, the amount of shape and motion detail which can be obtained depends on the resolution and number of examples. Enlarging the dataset can partially alleviate these issues but it is time-consuming to create a complex model. In the case of deformable models with defined deformation operators, the model design is a manual task requiring a good understanding of facial dynamics.

Model-free methods for dense facial tracking do not restrict the captured motion by a prior model but they are less robust to weak skin texture and fast non-rigid motions. This problem is usually circumvented by enhancing the skin texture with sparse *markers* or dense *pattern*. A sparse set of markers provides accurate alignment at their positions but surface points in between have approximate temporal consistency. A dense random pattern allows higher resolution of motion estimation which yields temporal alignment of detail such as skin wrinkling. Limiting factors for using the markers or pattern are inconvenience of their application on the face and occlusion of skin appearance.

To avoid uniformity of skin appearance, some approaches perform *geometry-based alignment* of per-frame 3D reconstructions without using image information. A drawback is the level of shape detail in geometries which amounts to medium-sized wrinkles and skin folds. Therefore, there is a limit to accuracy of the alignment working with the features of this scale. Moreover, large regions of the face are fairly smooth at the majority of frames which leads to drift of the tracked mesh.

With increasing camera resolution, many techniques pursue *image-based alignment* of the facial performance without use of markers or a pattern. Surface tracking leverages the large amount of skin detail present in images. Fine skin structure calculated by some approaches can augment the original image information to further increase the amount of detail. Despite the high image resolution and the augmentation with fine

skin geometry, concatenation of frame-to-frame alignments inevitably leads to accumulation of errors. Sequential tracking cannot also recover from a complete failure during complicated motions. The drift and failure problems are addressed by non-sequential tracking of the performance according to similarity between frames. However, the current methods use fairly simple non-sequential traversal of frames by jumping to neutral expressions occurring during the performance.

2.2 Photometric stereo

This section provides a brief overview of photometric stereo techniques which are useful for 3D capture of facial detail. Photometric stereo recovers normals and reflectance properties of surface points by analysing their image appearance under different lighting conditions. The approach works locally on a per-pixel basis, so fine geometric details visible in images can be reconstructed (e.g. skin wrinkles or pores).

Woodham [120] proposed the original *photometric stereo with white lights* (PSWL) for Lambertian surfaces which allows a linear formulation of the problem. An object is separately illuminated by three white lights from different known directions. Each light defines one linear constraint on the surface normal given the observed pixel intensity. Three gray-scale images from the same viewpoint provide enough constraints to determine the normal and gray-scale albedo at every pixel. In practice, pixel intensities can be corrupted by image noise, shadows, specular highlights, inter-reflections or subsurface scattering of light. These can bias the normal and albedo estimation but use of more lights solves some of these problems. Barsky and Petrou [8] employ 4 known light sources to detect shadows and specular highlights in individual images. A corrupted pixel in one image still allows correct estimation from three remaining images. This technique uses colour images which enables estimation of colour albedo for every pixel. Non-Lambertian surfaces are tackled by a higher number of lighting conditions [50, 43], so that shape and spatially-varying BRDF of multiple materials can be estimated across the surface.

Standard photometric stereo with time-multiplexing of lights is not suitable for dynamic capture due to surface motion between observations. An engineering solution is

high-speed image acquisition and light switching which requires specialised hardware to achieve satisfactory temporal sampling of the surface [94]. Image alignment is also necessary between individual observations to maintain accuracy of the result. Furthermore, a common drawback of photometric stereo methods is sensitivity to photometric calibration errors which cause low-frequency bias in estimated normals. Nehab et al. [77] eliminate this bias using a coarse shape by other 3D reconstruction techniques.

2.2.1 Photometric stereo with colour lights

Photometric stereo with colour lights separates illumination conditions spectrally rather than in time. Therefore, all illumination conditions can be recorded simultaneously in a single colour image. This is the crucial property for dynamic capture because normal maps of the surface can be reconstructed at every frame with no motion limitation. The initial concept was presented by Drew [33] which considers a Lambertian surface with unknown colour albedo and three spectrally different lights with unknown directions. A pixel colour changes only with the orientation of a surface normal assuming uniform colour albedo of the surface. Therefore, a single linear mapping between normals and pixel colours can be derived for the whole surface. It is possible to estimate the mapping from a single image up to an unknown global rotation of the normals.

Hernandez et al. [47, 20] recover the exact mapping for a given object from example normal-colour pairs by least-squares fitting. The example pairs are obtained by capturing a calibration board with a flat patch of the object material at various orientations. Normal orientation is fully calculated but material and light properties are not explicitly determined (included in the normal-colour mapping). The constant albedo assumption is alleviated to uniform surface chromaticity and varying grey-scale albedo in [48]. Subsequently, Hernandez et al. [46] propose self-calibration by capturing a short sequence of the object undergoing rigid motion. The normal-colour mapping is optimised for dominant chromaticity on the surface. Vogiatzis et al. [110] incorporate the Phong reflectance model which allows normal map computation for surfaces with mixed diffuse and specular reflectance. Monochromatic specular albedo of the surface is assumed to be constant together with the chromaticity.

The assumption of uniform chromaticity is restrictive for objects such as faces where normals in regions with non-dominant chromaticities would be biased. Anderson et al. [4] extend PSCL to Lambertian surfaces with multiple piece-wise constant chromaticities. This is achieved using a coarse normal map from stereo 3D reconstruction. Kim et al. [62] eliminate the uniform chromaticity assumption by resorting to time-multiplexing of three mixed colour illuminations. This enables estimation of normal and colour albedo maps of moving object at a half frame-rate of the camera. The captured images need to be aligned by optic flow but pixel accuracy is not required. Fyffe et al. [40] obtain per-pixel normal and colour albedo of the surface from a single image without time-multiplexing illumination. This is achieved by constructing camera system with 6 colour channels (bright and dark RGB). A multi-spectral pixel colour provides enough constraints for 5 degrees of freedom in the normal and colour albedo.

2.2.2 Photometric stereo with gradient illumination

Photometric stereo with gradient illumination has been developed in the context of facial capture with the Light Stage, thus some information are mentioned in Section 2.1.1. Gradient illumination avoids self-shadowing of the surface which is a problem of directional lights in PSWL and PSCL. Also, the light reaches the surface from all directions and is specularly reflected across the whole surface. This provides an option to separate diffuse and specular component of the reflection and compute dense specular reflectance data as well. The limitations are complex lighting setup and multiple observations under different gradient patterns. This requires high-speed recording and illumination switching for capturing dynamic objects.

Ma et al. [67] proposed original PSGI with three gradient and one constant illumination pattern. These patterns are captured twice under linearly polarised light to separate diffuse and specular reflection from the surface. Normals and albedos varying across the surface are computed for both reflection components. Efficient normal calculation by Vlasic et al. [109] adds three complement gradient illumination patterns. This also improves accuracy of diffuse normals because corresponding pixels are well exposed in at least one image from each complementary pairs. Wilson et al. [119] use complement

gradient patterns combined with light polarisation. Instead of doubling the number of observations only one additional image is necessary to approximately extract specular normal and albedo map.

2.2.3 Error analysis

Photometric stereo is susceptible to several kinds of error in the capture process which have an impact on accuracy of normals and albedos. Error analysis published in this area is focused on PSWL. Initial work by Ray et al. [89] compiles a comprehensive list of potential sources of imprecision and identify two major factors - image noise and calibration error in the light directions. Formulas describing the sensitivity of estimated surface gradient with respect to pixel intensity and light direction errors are derived. Jiang and Bunke [58] simplify this formulation by expressing surface orientation as a unit normal. Spence and Chantler [100] search for the optimal configuration of three light sources which minimises error in the estimated normal in the presence of image noise. Positions of the lights are optimised to decrease the ratio between uncertainty of the normals and the measured intensities which results in orthogonal light directions. Sun et al. [102] also confirm that the orthogonal light configuration minimises the impact of image noise and note that the normal error is largely dependent on surface albedo. Drbohlav and Chantler [32] theoretically derive the optimality of the orthogonal set of three lights and provide the optimal configuration for more than three lights. Multiple lights should be placed equidistantly on a circle with uniform slant $\sim 54.74°$. Barsky and Petrou [9] focus error analysis on shadow and specular highlight detection in their method [8].

2.3 Dense motion capture

This section provides a brief overview of dense motion capture which is necessary for achieving temporal consistency of facial performance capture. The focus is on image-based motion capture for non-rigid surfaces observed by multiple cameras.

2.3.1 Frame-to-frame scene flow

Dense motion capture of non-rigid surfaces was initially approached as independent estimation of 3D motion fields between consecutive frames. Vedula et al. [105, 104] introduced *scene flow* - a 3D vector field describing motion of the surface between two frames. They estimate the scene flow by fusion of 2D flow fields from multiple views for a volumetric model of the object. This requires pre-computation of 2D optic flows and 3D geometry at every frame. Latter work [106] does not require the geometry beforehand but works only for moving parts of the surface. Accuracy of the scene flow can be improved by incorporating error statistics of input optic flow and shape estimates [65]. Zhang et al. [125] integrate the scene flow estimation from 2D optic flows with MVS reconstruction.

Other methods estimate the scene flow directly without pre-computing 2D optic flows. They typically use a variational formulation of matching image information across views and between consecutive frames. Pons et al. [86] alternate between MVS and scene flow estimation in the same framework using a global image-based matching score. The shape and 3D motion of an object are discretised over a volumetric level set. Several approaches [126, 73, 56, 113] address calculation of a *disparity flow* which is a reduced definition of the scene flow for the binocular case. The 2D flow field and disparity change map are optimised over the reference image in the stereo pair. The approaches differ in construction of the energy functional in the variational framework.

The result of these techniques is a sequence of 3D shapes and instantaneous scene flows for the moving surface. There is no explicit temporal consistency in these data because the shape and motion are typically sampled at each frame over a regular grid in the 3D space or image domain. This is not suitable for facial performance capture which requires temporally consistent 3D models of the moving face.

2.3.2 Sequential surface tracking

Dense motion capture can use a common surface model for an object. The model is deformed over time by surface tracking according to multi-view image sequences. The

tracking is conventionally performed sequentially and scene flows between consecutive frames are calculated for control points of the deformed model. This brings temporal consistency into the resulting sequence of model instances.

A more tractable approach is to compute per-frame 3D geometries and 2D optic flows beforehand. A template mesh is deformed to fit the sequence of unaligned geometries and multi-view optic flows. Motion of vertices is optimised jointly subject to constraints given by the optic flows and raw geometries. Scene flows are usually regularised by enforcing smooth mesh deformation which suppresses incorrect motion vectors. This type of tracking has been demonstrated for a single camera [47, 119] where a 2.5D template mesh is warped between two frames by an optic flow and then back-projected onto a depth map. To reduce accumulation of errors in the optic flows, Wilson et al. [119] regularise the mesh warping by a local rigidity term.

Multi-camera tracking is presented by Zhang et al. [122] where the mesh deformation is constrained using 3D shapes by MVS. Regularisation of vertex motion is included in the scene flow estimation. In contrast, Bradley et al. [19] estimate a raw scene flow between two frames and then regularise it by Laplacian deformation. Despite various regularisation schemes, errors in frame-to-frame optic flows are propagated to the estimated 3D motion which leads to drift of the mesh. Bradley et al. [19] address the drift by additional optic flow estimation in the UV domain of the mesh after the initial deformation. The residual flow corrects the final positions of the vertices. However, the results are not satisfactory in regions undergoing fast and complex motion.

A disadvantage of the previous methods is pre-computation of 3D shapes and optic flows at all frames. Also, independent optic flow computation in every view results in inconsistencies of 2D flows which decrease accuracy of the resulting scene flow. *3D tracking* approaches overcome these disadvantages by joint estimation of shape and motion directly in the 3D space. Courchay et al. [26] extend the variational framework by Pons et al. [86] with mesh representation of the surface. They merge MVS and scene flow estimation to a single energy functional which is optimised across the whole mesh and over a temporal window. This results in a large optimisation which is computationally expensive and susceptible to local minima. To achieve more tractable

optimisation, the scene flow can be estimated using relatively small 3D patches attached to the surface. The surface patches can be completely independent and produce 3D trajectories for a sparse set of surface points [23, 29]. Another option is to associate the patches with control points of the surface model to deform it over time [79, 36].

Carceroni and Kutulakos [23] propose a comprehensive representation for the surface patches which consists of 3D position and orientation, curvature coefficients, diffuse/specular reflectance and linear transformation over time. This results in a complex optimisation scheme with a high number of parameters. A simpler approach [29] extends the Lucas-Kanade 2D tracking algorithm [5] to the 3D domain which aligns a textured planar surfel with images from multiple cameras. The surfel has a single texture template which has limited update over time to minimise the risk of drift. Neumann and Aloimonos [79] use a multi-resolution subdivision model for the surface instead of a collection of standalone patches. Deformation of the subdivision model between frames is iteratively refined by shape and motion optimisation of surface patches around its control points. Furukawa and Ponce [36] associate the patches with triangle fans around vertices of a surface mesh. The textured patches are tracked independently between two frames and raw motion vectors update vertex positions. Afterwards, the whole mesh is regularised by Laplacian smoothing combined with local mesh rigidity. The patch textures have fixed appearance from the reference frame which limits drift of the mesh over time.

A common problem of these approaches is inaccurate motion estimation for surfaces with weak and time-varying texture such as skin. Good temporal alignment of facial performances is shown only for faces with a painted pattern [36] or for high camera resolutions [19, 119]. However, significant drift of the surface model still appears after fast non-rigid motions and longer periods of time.

2.3.3 Non-sequential surface tracking

Recently, surface tracking methods have tackled the drift problem by *non-sequential* traversal of the input sequence. The template mesh is tracked from the initial frame along tree structure of paths leading to all frames. Shorter chains of frame-to-frame

alignments compared to sequential traversal reduce the amount of drift and the impact of a complete failure.

A simple approach is presented by Beeler et al. [13] for tracking facial performances. A traversal throughout a performance jumps directly from the initial frame to anchor frames with a similar neutral expression and then continues sequentially between them. Huang et al. [52] address global alignment of multiple un-registered mesh sequences of whole-body performances. At first, individual sequences are tracked separately by a geometry-based alignment method. Subsequently, frames from all sequences are compared in terms of shape dissimilarity of the original meshes. The sequences are linked through a few pairs of the least dissimilar frames which forms a sparsely connected graph among all frames. The shortest path tree is calculated on this graph which is weighted by frame-to-frame shape dissimilarities. The tree establishes a traversal according to which a template mesh is propagated to all frames using the temporal correspondences already computed. Budd et al. [22] construct a fully connected graph among frames in all sequences with edge weights given by the shape dissimilarity. The traversal is optimised by the minimum spanning tree which minimises the total path length through the dissimilarity space. In contrast to [52], this provides more optimal traversal tree which directly guides the actual surface tracking. Although, the tree-based non-sequential tracking reduces the drift, it suffers from alignment inconsistencies where different tree branches meet.

2.4 Conclusion

Review of the related work hints at potential research directions in the area of facial performance capture. This section highlights the most promising approaches with respect to the objectives of this work.

From the perspective of realism of the 4D performance model, capture methods using photometric stereo with colour lights offer a balanced solution for obtaining accurate pore-level skin detail [20, 110]. A capture setup from standard video equipment with no active illumination provides high-detail normal maps at camera frame-rate. On the other hand, appearance acquisition is limited but the focus of this work is on

facial geometry. The problem with low-frequency bias in photometric normals can be overcome by the combination with multi-view stereo 3D reconstruction. Multi-view stereo provides the correct medium-scale shape of the face at every frame which complements well the photometric stereo with colour lights. Stereo matching cannot be aided with a strong random pattern which would interfere with the detail acquisition. However, skin texture captured at high resolution enables accurate shape reconstruction without any pattern [19].

From the perspective of temporal consistency of the 4D performance model, the most practical methods calculate temporal alignment from natural facial appearance in multi-view image sequences. The challenge is accurate tracking of the skin during fast non-rigid facial movements without use of markers, pattern or prior deformable model. The majority of methods [122, 19, 119, 13] sequentially deform a template mesh of the face according to 2D optic flows in individual views and unregistered geometries. Disadvantages of this approach are pre-computation of the 3D geometry and flow fields at each frame and inconsistency of independent optic flow estimation across views. In contrast, 3D tracking techniques such as [36, 37] directly compute shape and motion of the mesh in the 3D space between consecutive frames.

Drift of the tracked mesh over time is the main limitation of recent capture systems for facial performances. This has been tackled by high-resolution capture to acquire more skin texture [19]. Another approach [119] is to use fine skin geometric detail together with image information for mesh tracking. These advances improve frame-to-frame temporal alignment but do not eliminate sequential accumulation of errors over longer periods of time. Non-sequential tracking proposed for whole-body performance capture [52, 22] offers an interesting mechanism to reduce the drift.

Chapter 3

Geometric detail capture

To create a realistic digital double of an actor it is crucial to capture the finest nuances of their performance. The system for facial performance capture has to be able to obtain a time-varying representation of facial shape up to fine skin structure. Our focus is on the geometry of skin detail rather than its appearance because it is much more difficult to manually model and animate believable dynamics of skin deformation than create a realistic skin texture.

Photometric stereo methods are suitable for reconstructing fine surface geometry for a wide range of materials. They estimate surface normals on a per-pixel basis which allows recovery of all detail visible in the input images. The only limit on scale of the obtained geometry is the resolution of the cameras used. Standard photometric stereo recovers normals of surface points by analysing their appearance under different directional illumination [120]. This is not suitable for analysing fast-moving objects such as a human face because it requires time-multiplexing of lighting conditions. The issue with surface motion between measurements can be circumvented by high frame-rate image acquisition with fast switching of light sources but requires specialised hardware [94].

Photometric stereo with gradient illumination [67] allows reconstruction of separate normals from the diffuse and specular component of light reflection but also requires switching between different illumination patterns. The reconstructed normal maps provide a great amount of skin detail for facial performance capture [69], but the main

disadvantage is the complex capture setup with high-speed cameras, projectors and a light stage. Photometric stereo with colour lights provides an alternative with a relatively simple light setup and standard cameras [20, 110]. Individual directional lights are separated spectrally rather than in time, thus all lighting conditions can be recorded simultaneously in a single colour image. Rapid actor motion does not pose a problem because normal maps are reconstructed independently for each frame. Approaches using shape-from-shading principle [14, 10] recover geometric detail from a single image but the normals are not metrically correct as in the photometric stereo.

In this chapter photometric stereo with colour lights is assessed in the context of detail capture for facial performance. A formulation of the classic photometric stereo with white lights is extended to colour illumination. Calibration for light directions and interaction between illumination, camera sensors and surface material is explained. A novel error analysis of this photometric technique is presented in terms of theoretical formulation and simulation on synthetic data. Finally, evaluation on real face data from performance capture is performed.

3.1 Photometric stereo with white lights

Following the work by Barsky and Petrou [8], photometric stereo with time-multiplexed white lights (PSWL) is based on several assumptions. The observed surface is assumed to be Lambertian resulting in a linear dependency between the observed intensity of an image pixel and the associated normal. To preserve this linearity, the camera sensor must have a linear response to incoming radiance. Moreover, the lights are modelled as point light source at a large distance which guarantees constant direction of light rays across the capture volume.

The grey-scale images $\{I_j\}_{j=1}^{J}$ of the surface are taken from the same viewpoint. The image I_j captures the surface illuminated by the light j with a direction vector $\mathbf{l}_j$ ($|\mathbf{l}_j| = 1$). Equation 3.1 describes the relationship between the pixel intensity g_j in I_j for a particular surface point and its corresponding normal $\mathbf{n}$.

$$g_j = \mathbf{l}_j^T \mathbf{n} \int E(\lambda) S(\lambda) R(\lambda) d\lambda = \mathbf{l}_j^T \alpha \mathbf{n} \tag{3.1}$$

The interaction between the light source, surface material and camera sensor is expressed by the integral over wavelength λ. The function $E(\lambda)$ is the light spectrum which is assumed to be the same for all light sources. The function $S(\lambda)$ is the spectral sensitivity of the grey-scale camera sensor. The function $R(\lambda)$ is the reflectance of the material which varies across the surface. The integral is represented by a scalar factor α which is commonly referred to as an *albedo* [8]. This expresses the appearance of the surface point with the given material in the camera sensor used which is independent from the spatial relationship between the surface and the light sources. However, the true surface albedo is different because it depends solely on the material reflectance $R(\lambda)$. The final intensity g_j is obtained by scaling α with the Lambertian dot product between $\mathbf{l}_j$ and $\mathbf{n}$.

For a stationary surface and camera, the intensities g_j of the same pixel across $\{I_j\}_{j=1}^{J}$ are measurements for the same surface point. This allows straightforward combination of constraints from individual images to estimate a per-pixel normal and albedo. Equation 3.2 defines a linear system constructed for every pixel by stacking Equation 3.1 across $\{I_j\}_{j=1}^{J}$. A vector $\mathbf{g}$ contains the image intensities g_j and a $3 \times J$ *illumination matrix* $\mathbf{L}$ consists of the light direction vectors $\mathbf{l}_j$.

$$\mathbf{g} = \mathbf{L}\alpha\mathbf{n} \quad \rightarrow \quad \begin{bmatrix} c_1 \\ \vdots \\ c_J \end{bmatrix} = \begin{bmatrix} \mathbf{l}_1^T \\ \vdots \\ \mathbf{l}_J^T \end{bmatrix} \alpha\mathbf{n} \tag{3.2}$$

In the case of 3 lights, a single solution is calculated by inversion of the known illumination matrix (Equation 3.3). Note that for $\mathbf{L}$ to be invertible, $\mathbf{l}_j$ have to be linearly independent (they should not lie on the same 3D plane). Because $\mathbf{n}$ is a unit vector, the albedo α equals $|\mathbf{L}^{-1}\mathbf{g}|$. In the case of more than 3 lights, the linear system is overdetermined and is solved in a least squares manner.

$$\alpha\mathbf{n} = \begin{cases} \mathbf{L}^{-1}\mathbf{g}, & J = 3 \\ (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{g}, & J > 3 \end{cases} \tag{3.3}$$

Barsky and Petrou [8] extended PSWL formulated above to colour images. The RGB

triples of a pixel in images $\{I_j\}_{j=1}^{J}$ should form a line in RGB space. However, Principal component analysis (PCA) is used to find its principal direction due to image noise. The direction is given by a chromaticity of the corresponding surface point. The grey-scale intensities g_j are obtained by projection of the RGB triplets onto the line. Afterwards, the normal $\mathbf{n}$ and albedo α can be calculated as in Equation 3.3. The benefit of this method is that appearance of the surface is described by colour albedo on a per-pixel basis by scaling colour direction with α.

3.2 Photometric stereo with colour lights

The main drawback of PSWL is time-multiplexing of lights which requires the camera and surface to be stationary until all illumination conditions are captured. Any motion of either of them causes errors in the estimated normal and albedo. This issue is addressed by photometric stereo based on colour lights (PSCL) [20]. Different lighting conditions are separated in wavelength rather than time. Therefore, the surface can be illuminated by all lights simultaneously. The number of lighting conditions is effectively limited to three by the number of sensors in a conventional colour camera (red, green, blue). A single colour image contains all the information necessary for the reconstruction of normal and albedo map. Therefore, this technique is suitable for moving surfaces because the surface detail can be reconstructed at each frame independently.

A single colour image I of the surface is captured under simultaneous illumination by spectrally different colour lights $j \in [1..3]$. Because of the independent reflection of different lights incident upon the surface, an observed RGB colour $\mathbf{c}$ of a surface point is the sum of contributions from the individual lights. This preserves the linearity of photometric calculation defined for PSWL in Equation 3.1. Equation 3.4 defines an observed intensity in the red channel c_r as the sum of intensities contributed by 3 lights with different spectra E_j.

$$c_r = \sum_{j=1}^{3} \mathbf{l}_j^T \mathbf{n} \int E_j(\lambda) S_r(\lambda) R(\lambda) d\lambda \tag{3.4}$$

The red camera sensor has its own spectral sensitivity S_r different from the green

and blue sensor. Each combination of sensor and light has a specific interaction, thus the integral is not constant for different light conditions as for PSWL (Equation 3.1). Therefore, the term albedo α from PSWL needs to be reformulated in the context of PSCL. Surface reflectance can be separated into wavelength-dependent chromaticity $\rho(\lambda)$ and spatially varying grey-scale albedo a ($R(\lambda) = a\rho(\lambda)$). In Equation 3.5 the albedo a is factored out of the integral and scales the unit normal $\mathbf{n}$.

$$c_r = \sum_{j=1}^{3} \mathbf{l}_j^T a\mathbf{n} \int E_j(\lambda) S_r(\lambda) \rho(\lambda) d\lambda \tag{3.5}$$

To simplify the formulation, Equation 3.6 defines a coefficient v_{rj} which encapsulates the integral from Equation 3.5 (similarly for the green and blue sensor).

$$v_{rj} = \int E_j(\lambda) S_r(\lambda) \rho(\lambda) d\lambda \tag{3.6}$$

A vector $\mathbf{v}_j = [v_{rj}\, v_{gj}\, v_{bj}]^T$ is then a response of the red, green and blue sensors to the light j given the surface material. Equation 3.7 shows a full relationship of the RGB colour $\mathbf{c}$ with the albedo-scaled normal $a\mathbf{n}$ (later referred to as a scaled normal).

$$\mathbf{c} = \mathbf{V}\mathbf{L}a\mathbf{n} \quad \rightarrow \quad \begin{bmatrix} c_r \\ c_g \\ c_b \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix} \begin{bmatrix} \mathbf{l}_1^T \\ \mathbf{l}_2^T \\ \mathbf{l}_3^T \end{bmatrix} a\mathbf{n} \tag{3.7}$$

The illumination matrix $\mathbf{L}$ has the size 3×3 and $\mathbf{V}$ denotes a 3×3 *interaction matrix* which consists of $\mathbf{v}_j$ for individual lights. A unique solution for the normal and albedo is calculated according to Equation 3.8 if $\mathbf{V}$ and $\mathbf{L}$ are known. Note that the vectors $\mathbf{v}_j$ cannot be linearly dependent so that $\mathbf{V}$ is invertible (similar condition as for $\mathbf{L}$ in Equation 3.3).

$$a\mathbf{n} = \mathbf{L}^{-1}\mathbf{V}^{-1}\mathbf{c} \tag{3.8}$$

An important assumption of the technique is that the chromaticity $\rho(\lambda)$ is uniform across the surface, thus one matrix $\mathbf{V}$ can be estimated for the whole surface. If $\rho(\lambda)$ varies across the surface, $\mathbf{V}$ would be an additional unknown for each pixel which leads to an under-constrained problem. The *uniform chromaticity assumption* allows

independent per-pixel estimation of $a\mathbf{n}$ using Equation 3.8 which yields a normal and grey-scale albedo map for the single image I. The presented formulation of PSCL allows correct calculation of the scaled normals only for surfaces with uniform chromaticity.

3.3 Calibration

The purpose of calibration is to determine the illumination matrix $\mathbf{L}$ and interaction matrix $\mathbf{V}$. The calibration of $\mathbf{L}$ means determining the light directions $\mathbf{l}_j$ with respect to the world coordinate system (WCS) in which the surface normals are expressed. The technique used is inspired by the work of Zhou and Kambhamettu [129]. A white specular sphere with known radius is placed at the centre of capture volume. The 3D location of the sphere is calculated from the centres of its projection in all views. The specular highlights on the sphere indicate the directions towards individual light sources. The ray back-projected through user-located image coordinates of specular highlight is reflected according to the normal at the point of incidence on the sphere. The reflected rays calculated from the highlights in all views for the light j are averaged to obtain a robust solution for $\mathbf{l}_j$. The resulting $\mathbf{L}$ is constant across the capture volume because the light sources are assumed to be distant and directional.

The technique by Hernandez et al. [48] has been modified for calibration of the interaction matrix $\mathbf{V}$ which is specific for the object captured. A calibration image $\bar{I}_j$ of the object is acquired under each colour light j separately (Figure 3.1(a-c)). The object is assumed stationary, so that the same pixel $\mathbf{p}$ in all three images corresponds with the same surface point with the scaled normal $a\mathbf{n}$. In the case of illumination by the single light j, a colour $\mathbf{c}$ of the pixel $\mathbf{p}$ in $\bar{I}_j$ is given by Equation 3.9 which is reduced from Equation 3.7.

$$\mathbf{c} = \mathbf{v}_j(\mathbf{l}_j^T a\mathbf{n}) \tag{3.9}$$

The vector $\mathbf{v}_j$ from $\mathbf{V}$ cannot be exactly calculated because $a\mathbf{n}$ is unknown during the calibration. However, a direction of $\mathbf{v}_j$ is given by the vector $\mathbf{c}$ which is a scaled version of $\mathbf{v}_j$. Equation 3.10 shows estimation of the direction over multiple pixels in $\bar{I}_j$ which

has robustness against image noise and varying surface chromaticity.

$$\sum_{\mathbf{p}\in P} \mathbf{c}_p = \mathbf{v}_j(\mathbf{l}_j^T \sum_{\mathbf{p}\in P} a_p \mathbf{n}_p) \tag{3.10}$$

The set of used pixels is defined by a binary mask P which is the same for all images $\bar{I}_j$ (Figure 3.1(d)). The mask P segments the maximal part of the object with dominant chromaticity which is not shadowed (provided by a user). A colour $\mathbf{c}_p$ of the pixel $\mathbf{p}$ is observed for a scaled normal $a_p\mathbf{n}_p$. The direction of $\mathbf{v}_j$ is obtained by normalising the accumulated vector $\sum_{\mathbf{p}\in P} \mathbf{c}_p$. This is repeated for every image $\bar{I}_j$ to form the full matrix $\mathbf{V}$.



(a) (b) (c) (d)

Figure 3.1: The image $\bar{I}_j$ of an actor with painted white make-up captured under each colour light j separately (a,b,c). A mask P marks unshadowed facial region covered with the make-up (d).

The object is illuminated by each light from the same direction during the calibration. This differs from the spatially distributed positions used for the actual normal estimation. The reason is that the term $\mathbf{l}_j^T \sum_{\mathbf{p}\in P} a_p \mathbf{n}_p$ from Equation 3.10 is unknown but the same for all images $\bar{I}_j$. Thus, ratios between magnitudes of $\mathbf{v}_j$ are equal to ratios between magnitudes of their $\sum_{\mathbf{p}\in P} \mathbf{c}_p$. After establishing relative ratios the final magnitudes are given by setting the largest $\mathbf{v}_j$ to a unit vector. This calculation provides more accurate $\mathbf{V}$ than Hernandez et al. [48] who illuminate the object by individual lights from different directions. They estimate the directions of $\mathbf{v}_j$ the same way but the final magnitudes of $\mathbf{v}_j$ are given by ratios between the magnitudes of the largest $\mathbf{c}_p$ under individual lights. This means that the strength of sensor response for a particular

light depends on a colour of the brightest pixel which is sensitive to image noise and specular highlights.

3.4 Error analysis

Quality of detail reconstruction using PSCL is influenced by image noise or errors in the parameters describing capture setup (illumination and interaction matrices). The error analysis published in the literature (Section 2.2.3) has been in the context of PSWL and does not consider the interaction matrix. The majority of previous work is focused on finding optimal light configuration in the presence of image noise [89, 58, 100, 32, 102]. There is limited analysis of small inaccuracies in light direction vectors for few example normal directions [89, 58]. The aim of this analysis is to understand how errors in the estimation of illumination and interaction matrices affect the reconstructed normals (secondarily albedos). Influence of image noise is also investigated to validate the previous research.

3.4.1 Error in the illumination matrix

The theoretical analysis of PSCL is initially performed with an arbitrary matrix $\mathbf{L}$ and the identity matrix $\mathbf{V}$ on an object with white uniform chromaticity (equivalent to PSWL with 3 lights of the same intensity). The simulation-reconstruction chain in Equation 3.11 firstly calculates a colour $\mathbf{c}$ for a scaled normal $\tilde{\mathbf{n}}_S = a_S\mathbf{n}_S$ using the actual parameters of a capture setup (index S). Secondly, the scaled normal $\tilde{\mathbf{n}}_R = a_R\mathbf{n}_R$ is reconstructed from $\mathbf{c}$ using estimated parameters (index R). The error between the original $\tilde{\mathbf{n}}_S$ and the reconstructed $\tilde{\mathbf{n}}_R$ is expressed as a difference vector $\mathbf{d}$.

$$\mathbf{c} = \mathbf{L}_S\tilde{\mathbf{n}}_S \quad \rightarrow \quad \tilde{\mathbf{n}}_R = \mathbf{L}_R^{-1}\mathbf{c} \quad \rightarrow \quad \mathbf{d} = \tilde{\mathbf{n}}_R - \tilde{\mathbf{n}}_S = \mathbf{L}_R^{-1}(\mathbf{L}_S - \mathbf{L}_R)\tilde{\mathbf{n}}_S \tag{3.11}$$

Assume a discrepancy in estimation of the blue light direction $\mathbf{l}_3$, thus the illumination matrices $\mathbf{L}_S$ and $\mathbf{L}_R$ have different rows $\mathbf{l}_{3S}^T$ and $\mathbf{l}_{3R}^T$. Figure 3.2(a) illustrates an example case where the discrepancy $(\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T)$ is in the tilt ϕ of blue light direction (the slant

θ is unchanged). Equation 3.12 expands the expression of $\mathbf{d}$ from Equation 3.11 and simplifies it for the case of error in $\mathbf{l}_3$. Equation 3.12 can be rearranged into Equation 3.13 to separate the direction of $\mathbf{d}$ (*term C*) and the magnitude of $\mathbf{d}$ (*term A*$\cdot$*term B*).

$$\mathbf{d} = \begin{bmatrix} \mathbf{l}_1^T \\ \mathbf{l}_2^T \\ \mathbf{l}_{3R}^T \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{l}_1^T \\ \mathbf{l}_2^T \\ \mathbf{l}_{3S}^T \end{bmatrix} - \begin{bmatrix} \mathbf{l}_1^T \\ \mathbf{l}_2^T \\ \mathbf{l}_{3R}^T \end{bmatrix} \right) \tilde{\mathbf{n}}_S = \frac{1}{\mathbf{l}_{3R}^T(\mathbf{l}_1 \times \mathbf{l}_2)} (\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T) \tilde{\mathbf{n}}_S (\mathbf{l}_1 \times \mathbf{l}_2) \quad (3.12)$$

$$\mathbf{d} = \underbrace{\frac{1}{\cos\gamma_3}}_{term\ A} \underbrace{|\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T||\tilde{\mathbf{n}}_S| cos\beta_3}_{term\ B} \underbrace{\mathbf{m}_3}_{term\ C} \quad (3.13)$$

The vector $\mathbf{m}_3$ is the normalised vector $(\mathbf{l}_1 \times \mathbf{l}_2)$ and γ_3 is the angle between $\mathbf{m}_3$ and $\mathbf{l}_{3R}$. The *term A* is a scalar coefficient which scales $\mathbf{d}$ according to the spatial configuration of lights. The *term B* defines a zero-error plane whose normal is given by the discrepancy $(\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T)$ and which always crosses the origin of WCS (illustrated in Figure 3.2(a)). On this 3D plane $|\mathbf{d}|$ vanishes because the angle β_3 between $\tilde{\mathbf{n}}_S$ and $(\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T)$ becomes $90°$ ($cos\beta_3 = 0$). The magnitude $|\mathbf{d}|$ linearly increases outside the zero-error plane with the distance between the tip of $\tilde{\mathbf{n}}_S$ and the plane which is expressed by $|\tilde{\mathbf{n}}_S| cos\beta_3$. The slope of this dependency is given by the magnitude of discrepancy $|\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T|$. The *term B* also flips the direction of $\mathbf{d}$ depending which side of the zero-error plane $\tilde{\mathbf{n}}_S$ is on (illustrated by vectors $\mathbf{d}, \mathbf{d}'$ in Figure 3.2).

Equation 3.14 expands Equation 3.11 for the general case where there are discrepancies in all light directions.

$$\mathbf{d} = \begin{bmatrix} \mathbf{l}_{1R}^T \\ \mathbf{l}_{2R}^T \\ \mathbf{l}_{3R}^T \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{l}_{1S}^T \\ \mathbf{l}_{2S}^T \\ \mathbf{l}_{3S}^T \end{bmatrix} - \begin{bmatrix} \mathbf{l}_{1R}^T \\ \mathbf{l}_{2R}^T \\ \mathbf{l}_{3R}^T \end{bmatrix} \right) \tilde{\mathbf{n}}_S = \mathbf{d}_1 + \mathbf{d}_2 + \mathbf{d}_3 \quad (3.14)$$

$$\mathbf{d}_1 = \frac{1}{\mathbf{l}_{1R}^T(\mathbf{l}_{2R} \times \mathbf{l}_{3R})} (\mathbf{l}_{1S}^T - \mathbf{l}_{1R}^T) \tilde{\mathbf{n}}_S (\mathbf{l}_{2R} \times \mathbf{l}_{3R}) = \frac{1}{\cos\gamma_1} |\mathbf{l}_{1S}^T - \mathbf{l}_{1R}^T||\tilde{\mathbf{n}}_S| cos\beta_1 \mathbf{m}_1$$

$$\mathbf{d}_2 = \frac{1}{\mathbf{l}_{2R}^T(\mathbf{l}_{3R} \times \mathbf{l}_{1R})} (\mathbf{l}_{2S}^T - \mathbf{l}_{2R}^T) \tilde{\mathbf{n}}_S (\mathbf{l}_{3R} \times \mathbf{l}_{1R}) = \frac{1}{\cos\gamma_2} |\mathbf{l}_{2S}^T - \mathbf{l}_{2R}^T||\tilde{\mathbf{n}}_S| cos\beta_2 \mathbf{m}_2$$

$$\mathbf{d}_3 = \frac{1}{\mathbf{l}_{3R}^T(\mathbf{l}_{1R} \times \mathbf{l}_{2R})} (\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T) \tilde{\mathbf{n}}_S (\mathbf{l}_{1R} \times \mathbf{l}_{2R}) = \frac{1}{\cos\gamma_3} |\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T||\tilde{\mathbf{n}}_S| cos\beta_3 \mathbf{m}_3$$

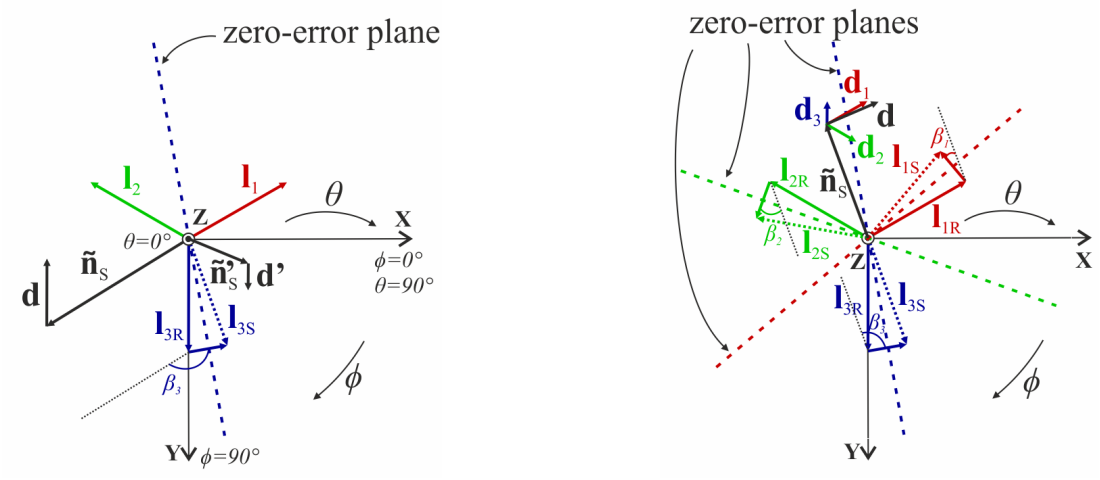


Figure 3.2: Spatial relationship between an error $\mathbf{d}$ for a scaled normal $\tilde{\mathbf{n}}_S$, actual light directions $\mathbf{L}_S$ and estimated light directions $\mathbf{L}_R$ in WCS (viewed against the $\mathbf{Z}$-axis). A discrepancy between $\mathbf{L}_S$, $\mathbf{L}_R$ in the tilt of blue light (a), in the tilt of all lights (b).

An important observation is that $\mathbf{d}$ is a vector sum of the difference vectors $\mathbf{d}_j$ introduced by errors in individual $\mathbf{l}_j$ (as defined by Equations 3.12, 3.13). Because of this combination the direction and magnitude of the resulting $\mathbf{d}$ can vary across the space of scaled normals in a complex way. This depends on the spatial configuration of light sources and location of the zero-error planes created by the respective discrepancies in their estimated direction. Figure 3.2(b) illustrates an example situation with all lights tilted in one direction by the same angle. The final difference vector $\mathbf{d}$ vanishes only for $\tilde{\mathbf{n}}_S$ aligned with the $\mathbf{Z}$-axis where the zero-error planes cross each other. Otherwise, its magnitude increases with distance from $\mathbf{Z}$-axis.

Note that $|\mathbf{d}|$ for any $\tilde{\mathbf{n}}_S$ directly depends on the spatial relationship between $\mathbf{l}_{jR}$. Each contribution $\mathbf{d}_j$ is scaled by $\frac{1}{\cos\gamma_j}$ (*term A* in Equation 3.13) which is independent of the magnitude of discrepancy. The *term A* is minimal for each light ($\frac{1}{\cos\gamma_j} = 1$) if $\mathbf{l}_{jR}$ is an orthonormal set of vectors. Therefore, there is no additional scaling of $\mathbf{d}$ due to positioning of lights when their directions are perpendicular to each other.

3.4.2 Error in the interaction matrix

The analysis in the previous section can be expanded to include discrepancies in an arbitrary interaction matrix $\mathbf{V}$. In this case it is assumed that only the estimation of $\mathbf{V}$ is incorrect ($\mathbf{V}_S \neq \mathbf{V}_R, \mathbf{L} = \mathbf{L}_S = \mathbf{L}_R$). The simulation-reconstruction chain in Equation 3.15 is therefore modified to reflect this.

$$\mathbf{c} = \mathbf{V}_S\mathbf{L}\tilde{\mathbf{n}}_S \rightarrow \tilde{\mathbf{n}}_R = \mathbf{L}^{-1}\mathbf{V}_R^{-1}\mathbf{c} \quad \rightarrow \quad \mathbf{d} = \tilde{\mathbf{n}}_R - \tilde{\mathbf{n}}_S = \mathbf{L}^{-1}\mathbf{V}_R^{-1}(\mathbf{V}_S - \mathbf{V}_R)\mathbf{L}\tilde{\mathbf{n}}_S \quad (3.15)$$

A formula for $\mathbf{d}$ with individual vector components is shown in Equation 3.16.

$$\mathbf{d} = \begin{bmatrix} \mathbf{l}_1^T \\ \mathbf{l}_2^T \\ \mathbf{l}_3^T \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{v}_{1R} & \mathbf{v}_{2R} & \mathbf{v}_{3R} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{v}_{1S} - \mathbf{v}_{1R} & \mathbf{v}_{2S} - \mathbf{v}_{2R} & \mathbf{v}_{3S} - \mathbf{v}_{3R} \end{bmatrix} \begin{bmatrix} \mathbf{l}_1^T \\ \mathbf{l}_2^T \\ \mathbf{l}_3^T \end{bmatrix} \tilde{\mathbf{n}}_S \quad (3.16)$$

By decomposition of matrix $(\mathbf{V}_S - \mathbf{V}_R)$ it is possible to separate error contributions made by discrepancies in individual vectors $\mathbf{v}_j$. This results in the sum $\mathbf{d} = \mathbf{d}_1 + \mathbf{d}_2 + \mathbf{d}_3$ as for the illumination matrix $\mathbf{L}$ in Equation 3.14. The difference vector $\mathbf{d}_1$ expressed in Equation 3.17 is related to a discrepancy in $\mathbf{v}_1$.

$$\begin{aligned} \mathbf{d}_1 = & \\ & \frac{1}{\mathbf{l}_1^T(\mathbf{l}_2 \times \mathbf{l}_3)} \begin{bmatrix} \mathbf{l}_2 \times \mathbf{l}_3 & \mathbf{l}_3 \times \mathbf{l}_1 & \mathbf{l}_1 \times \mathbf{l}_2 \end{bmatrix} \frac{1}{\mathbf{v}_{1R}^T(\mathbf{v}_{2R} \times \mathbf{v}_{3R})} \begin{bmatrix} (\mathbf{v}_{2R} \times \mathbf{v}_{3R})^T \\ (\mathbf{v}_{3R} \times \mathbf{v}_{1R})^T \\ (\mathbf{v}_{1R} \times \mathbf{v}_{2R})^T \end{bmatrix} (\mathbf{v}_{1S} - \mathbf{v}_{1R})\mathbf{l}_1\tilde{\mathbf{n}}_S \\ & = \underbrace{\begin{bmatrix} \frac{\mathbf{m}_1}{\cos\gamma_1} & \frac{\mathbf{m}_2}{\cos\gamma_2} & \frac{\mathbf{m}_3}{\cos\gamma_3} \end{bmatrix} \begin{bmatrix} \frac{\mathbf{u}_1^T}{|\mathbf{v}_{1R}|\cos\delta_1} \\ \frac{\mathbf{u}_2^T}{|\mathbf{v}_{2R}|\cos\delta_2} \\ \frac{\mathbf{u}_3^T}{|\mathbf{v}_{3R}|\cos\delta_3} \end{bmatrix}}_{term\ A} \underbrace{|\tilde{\mathbf{n}}_S|\cos\eta_1}_{term\ B} \underbrace{(\mathbf{v}_{1S} - \mathbf{v}_{1R})}_{term\ C} \end{aligned} \quad (3.17)$$

The magnitude $|\mathbf{d}_1|$ linearly increases from a zero-error 3D plane as for errors in the illumination matrix $\mathbf{L}$. The normal of the plane is $\mathbf{l}_1$ and *term B* defines the distance of $\tilde{\mathbf{n}}_S$ from it where η_1 is an angle between $\tilde{\mathbf{n}}_S$ and $\mathbf{l}_1$. The increase of $|\mathbf{d}_1|$ depends on the discrepancy vector (*term C*) but the final scaling coefficient is given by the magnitude

of this vector after transformation by the matrices in *term A*. The direction of $\mathbf{d}_1$ is also set by the transformed discrepancy vector and is the same for all $\tilde{\mathbf{n}}_S$ apart from opposite sign on each side of the zero-error plane. The matrix $\mathbf{V}_R^{-1}$ in *term A* can be expressed in terms of unit vectors $\mathbf{u}_j$ in a similar manner to $\mathbf{L}^{-1}$ in Equation 3.14 with the difference that the original $\mathbf{v}_j$ do not have to be unit vectors. The vector $\mathbf{u}_1$ is the normalised vector $\mathbf{v}_{2R} \times \mathbf{v}_{3R}$ and the angle δ_1 is defined between $\mathbf{u}_1$ and $\mathbf{v}_{1R}$ (respectively for other $\mathbf{u}_j$). Equation 3.18 for the aggregate error vector $\mathbf{d}$ reformulates Equation 3.16 according to inference in Equation 3.17.

$$\begin{aligned}\mathbf{d} &= \mathbf{d}_1 + \mathbf{d}_2 + \mathbf{d}_3 = \begin{bmatrix} \frac{\mathbf{m}_1}{\cos\gamma_1} & \frac{\mathbf{m}_2}{\cos\gamma_2} & \frac{\mathbf{m}_3}{\cos\gamma_3} \end{bmatrix} \begin{bmatrix} \frac{\mathbf{u}_1^T}{|\mathbf{v}_{1R}|\cos\delta_1} \\ \frac{\mathbf{u}_2^T}{|\mathbf{v}_{2R}|\cos\delta_2} \\ \frac{\mathbf{u}_3^T}{|\mathbf{v}_{3R}|\cos\delta_3} \end{bmatrix} \cdot \\ &(|\tilde{\mathbf{n}}_S|\cos\eta_1(\mathbf{v}_{1S} - \mathbf{v}_{1R}) + |\tilde{\mathbf{n}}_S|\cos\eta_2(\mathbf{v}_{2S} - \mathbf{v}_{2R}) + |\tilde{\mathbf{n}}_S|\cos\eta_3(\mathbf{v}_{3S} - \mathbf{v}_{3R}))\end{aligned} \tag{3.18}$$

The zero-error planes are perpendicular to the light directions $\mathbf{l}_j$ and intersect at the origin of WCS. Each discrepancy vector $(\mathbf{v}_{jS} - \mathbf{v}_{jR})$ is multiplied by common factor $\mathbf{L}^{-1}\mathbf{V}_R^{-1}$ (*term A* in Equation 3.17) and scaled by a distance from the respective zero-error plane (*term B* in Equation 3.17). As was mentioned in Section 3.4.1, the scaling coefficients $\frac{1}{\cos\gamma_j}$ are minimal when the light directions are orthogonal. In this case, the length of any vector is not changed by multiplication with $\mathbf{L}^{-1}$. Equivalently, coefficients $\frac{1}{|\mathbf{v}_{jR}|\cos\delta_j}$ in $\mathbf{V}_R^{-1}$ are minimal when $\mathbf{v}_{jR}$ is an orthogonal set of vectors and their magnitudes are large. Under these conditions for $\mathbf{V}_R$ and $\mathbf{L}$, $|\mathbf{d}|$ is scaled the least for any $\tilde{\mathbf{n}}_S$ regardless of the magnitude of discrepancies in the interaction matrix.

3.4.3 Image noise

The last source of imprecision in reconstructed normals is the image noise. Assume for this case that the estimation of both $\mathbf{V}$ and $\mathbf{L}$ is correct ($\mathbf{V} = \mathbf{V}_S = \mathbf{V}_R, \mathbf{L} = \mathbf{L}_S = \mathbf{L}_R$). The simulation-reconstruction chain is then defined as Equation 3.19 where $\boldsymbol{\Delta}$ is a noise vector from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$. Note that $\mathbf{d}$ does not depend on $\tilde{\mathbf{n}}_S$ as in other sources of errors.

$$\mathbf{c} = \mathbf{V}\mathbf{L}\tilde{\mathbf{n}}_S \rightarrow \tilde{\mathbf{n}}_R = \mathbf{L}^{-1}\mathbf{V}^{-1}(\mathbf{c} + \boldsymbol{\Delta}) \quad \rightarrow \quad \mathbf{d} = \tilde{\mathbf{n}}_R - \tilde{\mathbf{n}}_S = -\mathbf{L}^{-1}\mathbf{V}^{-1}\boldsymbol{\Delta} \tag{3.19}$$

Optimality of light directions with respect to the noise in intensities has been investigated in the context of PSWL [102, 100, 32]. The conclusion is that $\mathbf{L}$ has to be orthogonal to minimise the impact of image noise. The theoretical proof by Drbohlav and Chantler [32] can be generalised in a straightforward manner to include the interaction matrix $\mathbf{V}$ from the formulation for PSCL. As the result, the combined matrix $\mathbf{VL}$ has to be formed by orthogonal vectors. Assuming $\mathbf{L}$ to be orthonormal, $\mathbf{v}_j$ need to be perpendicular to each other as well. Note that entries in $\mathbf{V}$ are all positive (integrals over wavelength in Equation 3.6), therefore $\mathbf{V}$ needs to be diagonal to fulfil the orthogonality criterion. In practice, this means that the light spectra and spectral sensitivity of sensors should be matched ($E_1(\lambda) = S_r(\lambda), E_2(\lambda) = S_g(\lambda), E_3(\lambda) = S_b(\lambda)$).

3.4.4 Experiments

The presented theoretical formulations are investigated by a set of experiments on synthetic data. A white hemisphere is chosen as a test object in a virtual capture setup because it provides all possible normal directions visible from a single camera. The white albedo is uniform across the hemisphere to fulfil the chromaticity assumption ($|\tilde{\mathbf{n}}_S| = 255$). The virtual capture setup is illustrated in Figure 3.2. The hemisphere is placed at the origin of WCS and the camera points along the $\mathbf{Z}$-axis towards the origin. The light directions have the same slant $\theta = 26^\circ$ and are equidistantly spaced tilts $\phi_1 = 330^\circ$, $\phi_2 = 210^\circ$ and $\phi_3 = 90^\circ$. Note that this is not the optimal set of light directions, however it is similar to the real setup used for facial performance capture. The spectra of lights are matched with spectral sensitivity curves of the camera sensors ($\mathbf{V}$ is the identity matrix), thus red, green and blue light are used. A ground-truth normal and albedo map of the hemisphere is generated for the virtual camera assuming orthographic projection.

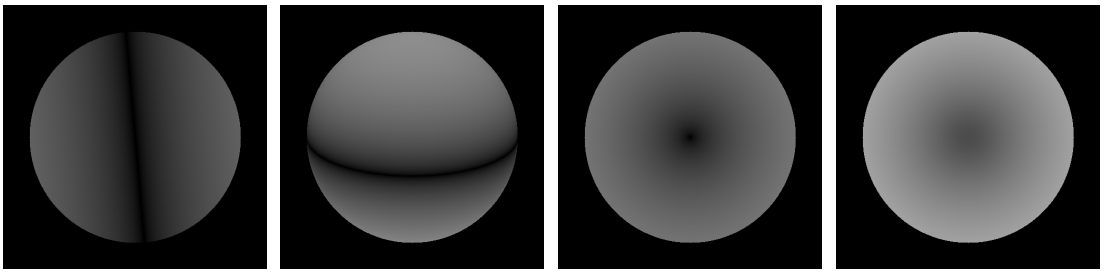
The simulation-reconstruction chain initially creates an image of the hemisphere from the ground-truth albedo-scaled normal map using actual parameters of the capture

setup $(\mathbf{V}_S, \mathbf{L}_S)$. Afterwards, the map is reconstructed back from the image using estimated parameters $(\mathbf{V}_R, \mathbf{L}_R)$. The matrices $\mathbf{V}_R$ and $\mathbf{L}_R$ correspond to the description of the capture setup because it is assumed that the system is built precisely according to the specification. The matrices $\mathbf{V}_S$ and $\mathbf{L}_S$ deviated from $\mathbf{V}_R$ and $\mathbf{L}_R$ represent the actual construction of the setup. The difference between the albedo-scaled normal maps is examined for different types of discrepancies. Patterns of $|\mathbf{d}|$ over the hemisphere show how the error vector spatially varies depending on the normal direction. Different root mean square (RMS) statistics across the whole hemisphere are plotted for increasing magnitude of individual discrepancies. They include RMS of the magnitude of the difference vector $|\mathbf{d}|$, RMS of the angle between actual and reconstructed normal $\angle(\tilde{\mathbf{n}}_S, \tilde{\mathbf{n}}_R)$ and RMS of the difference between actual albedo and reconstructed albedo $(a_S - a_R)$. The angle $\angle(\tilde{\mathbf{n}}_S, \tilde{\mathbf{n}}_R)$ and the difference $(a_S - a_R)$ are added to illustrate a split of $\mathbf{d}$ between the normal direction and the albedo. Note that shadows are not modelled during the simulation and negative RGB values in areas occluded from any light do not affect correct reconstruction of normals.

Error in the illumination matrix

The first set of experiments is focused on discrepancies in light directions and the simulation-reconstruction chain follows Equation 3.11 (note that $\mathbf{V}$ is fixed). In Experiment A the actual direction of blue light $\mathbf{l}_{3S}$ is tilted away from the estimated direction $\mathbf{l}_{3R}$ by an angle $\triangle\phi$ as shown in Figure 3.2(a). The pattern in Figure 3.3(a) corresponds to theoretical expression in Equation 3.13. The black belt across the hemisphere indicates the zero-error plane. The magnitude $|\mathbf{d}|$ increases linearly as normals tilt away from this plane. With changing $\triangle\phi$ the plane rotates because $(\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T)$ changes its direction. There is a linear dependency between $|\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T|$ and $|\mathbf{d}|$ according to Equation 3.13. However, the relationship between $|\mathbf{l}_{3S}^T - \mathbf{l}_{3R}^T|$ and $\triangle\phi$ is sinusoidal. Therefore, the RMS of $|\mathbf{d}|$ in Figure 3.4(a) increases sinusoidally but the trend is effectively linear for small angles. The magnitude $|\mathbf{d}|$ is symmetrical around $\triangle\theta = 0$, so the error does not depend on the direction of tilt. The graph also shows that the error in normal direction and albedo has a similar trend as the error in scaled normal. The RMS of $\angle(\tilde{\mathbf{n}}_S, \tilde{\mathbf{n}}_R)$ is lower than the introduced angular discrepancy in the light direction. The remaining

inaccuracy is propagated to the albedo which is proportionally more affected than the direction of normal. In Experiment B $\mathbf{l}_{3S}$ is slanted away from the estimated $\mathbf{l}_{3R}$ by an angle $\triangle\theta$. In comparison to Experiment A the position of the zero-error plane is different but $|\mathbf{d}|$ changes across the hemisphere in a similar way (Figure 3.3(b)). The RMS curves in Figure 3.4(b) depict more severe impact on the quality of reconstruction than for a tilt. This is due to spatial location of the zero-error plane with respect to the hemisphere.



(a) (b) (c) (d)

Figure 3.3: The magnitude of error in scaled normal $|\mathbf{d}|$ across the hemisphere: Exp. A - the change $\triangle\phi = -10^\circ$ in the tilt of blue light (a), Exp. B - the change $\triangle\theta = -10^\circ$ in the slant of blue light (b), Exp. C - the change $\triangle\phi = -10^\circ$ in the tilt of all lights (c), Exp. D - the change $\triangle\theta = -10^\circ$ in the slant of all lights (d). The magnitude $|\mathbf{d}|$ is encoded in grey-scale - black = 0, white = 255 coordinate units.

All light directions $\mathbf{l}_{jS}$ are tilted in the same direction by an angle $\triangle\phi$ in Experiment C as depicted in Figure 3.2(b). The error vector $\mathbf{d}$ is theoretically expressed in Equation 3.14. The pattern of $|\mathbf{d}|$ in Figure 3.3(c) shows that the only zero error is on top of the hemisphere and $|\mathbf{d}|$ increases radially from the $\mathbf{Z}$-axis. This reflects the theoretical formulation of three zero-error planes crossing each other along the $\mathbf{Z}$-axis. The graph in Figure 3.4(c) depicts that the dependency between the RMS of $|\mathbf{d}|$ and $\triangle\phi$ is again effectively linear for small angular discrepancies (reflecting Equation 3.14). Interestingly, the albedo error is zero regardless of $\triangle\phi$. This is caused by the fact that scaled normals only change direction opposite to the rotation of the light setup and their scale is not modified. In Experiment D all light directions $\mathbf{l}_{jS}$ are slanted in the same direction by an angle $\triangle\theta$. The zero-error planes for individual lights cut through the hemisphere in the same manner as in the single-light case (Experiment B). They intersect at the origin of WCS, therefore none of the normals in Figure 3.3(d) is completely accurate.

The magnitude $|\mathbf{d}|$ is smallest along the **Z**-axis and decreases radially towards the edges of the hemisphere. The graphs in Figure 3.4(b, d) show that a discrepancy in the slant angle generally has a stronger impact on the quality of the result than in the tilt angle. In all experiments apart from C the albedo has higher error compared to its maximal possible value than a deviation in the normal. Angular error of normal is always the same or smaller than the discrepancy in light directions.

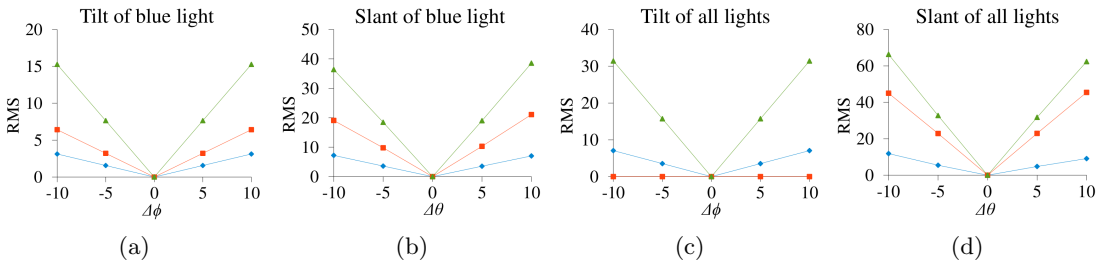


Figure 3.4: The RMS graphs across different magnitudes of discrepancy for Exp. A - D (a-d): RMS of $|\mathbf{d}|$ - green (in coordinate units), RMS of $(a_S - a_R)$ - red (in grey-scale levels) and RMS of $angle(\tilde{\mathbf{n}}_S, \tilde{\mathbf{n}}_R)$ - blue (in degrees).

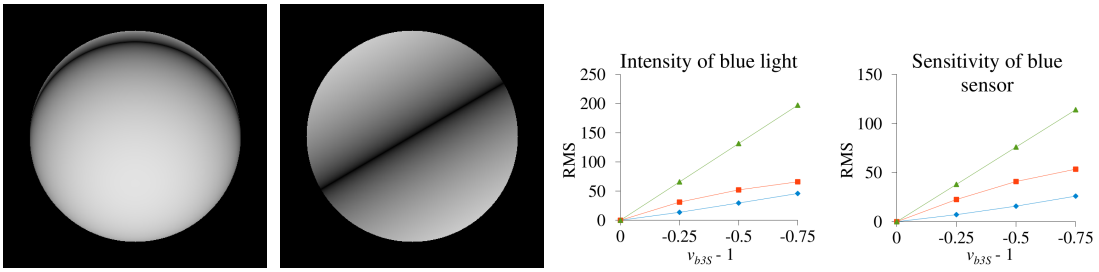
Error in the interaction matrix

The second set of experiments is focused on discrepancies in the interaction matrix $\mathbf{V}$ and the simulation-reconstruction chain follows Equation 3.15 (note that $\mathbf{L}$ is fixed). $\mathbf{V}_S$ is deviated in individual coefficients from the ideal identity matrix $\mathbf{V}_R$. In Experiment E only the coefficient v_{b3S} which describes the interaction between the blue light and blue camera sensor is changed. A decrease of v_{b3S} from the original value 1 simulates lowering intensity of the blue light. Equation 3.18 describing the general discrepancy between $\mathbf{V}_S$ and $\mathbf{V}_R$ has a simpler form of Equation 3.20 because only the contribution $\mathbf{d}_3$ for the blue light is non-zero.

$$\mathbf{d} = \frac{\mathbf{m}_3}{\cos\gamma_3}|\tilde{\mathbf{n}}_S|\cos\eta_3(v_{b3S} - 1) \tag{3.20}$$

Equation 3.20 is an equivalent of Equation 3.17 for the vector $\mathbf{v}_3$. The *term A* in Equation 3.17 is simplified to $\frac{\mathbf{m}_3}{\cos\gamma_3}$ because $\mathbf{V}_R^{-1}$ is the identity matrix and the discrepancy $(v_{b3S} - 1)$ selects only the last column from $\mathbf{L}^{-1}$. The normal of zero-error plane equals

$\mathbf{l}_3$ (stems from the term $\cos\eta_3$). The vector $|\mathbf{d}|$ linearly increases with distance from the plane as shown in Figure 3.5(a). The direction of $\mathbf{d}$ is given by $\mathbf{m}_3$ across whole hemisphere and does not change with $(v_{b3S}-1)$.



(a) (b) (c) (d)

Figure 3.5: The magnitude of error in scaled normal $|\mathbf{d}|$ across the hemisphere: Exp. E - intensity of blue light where $(v_{b3S}-1)=-0.5$ (a), Exp. F - sensitivity of blue sensor where $(v_{b3S}-1)=-0.5$(b). The RMS graphs across different magnitudes of $(v_{b3S}-1)$ for Exp. E (c) and Exp. F (d).

The coefficients v_{b2S} and v_{b3S} are changed simultaneously in Experiment F. Their sum remains equal to 1, thus the discrepancy is expressed as $(v_{b3S}-1)$ (as in Experiment E). This experiment simulates a blue sensor sensitive to both green and blue light which shifts the sensitivity curve from the blue to green light spectrum with decreasing v_{b3S}. The inference from Equation 3.18 simplifies the terms $\mathbf{d}_2$ and $\mathbf{d}_3$ in a similar way as for Equation 3.20. The combination of these two zero-error planes in Equation 3.21 creates a single zero-error plane due to the constraint $v_{b2S}+v_{b3S}=1$.

$$\begin{aligned}\mathbf{d} &= \frac{\mathbf{m}_3}{\cos\gamma_3}(|\tilde{\mathbf{n}}_S|\cos\eta_2(v_{b2S}-0)+|\tilde{\mathbf{n}}_S|\cos\eta_3(v_{b3S}-1))\\ \mathbf{d} &= \frac{\mathbf{m}_3}{\cos\gamma_3}|\tilde{\mathbf{n}}_S|(\cos\eta_3-\cos\eta_2)(v_{b3S}-1)\end{aligned} \tag{3.21}$$

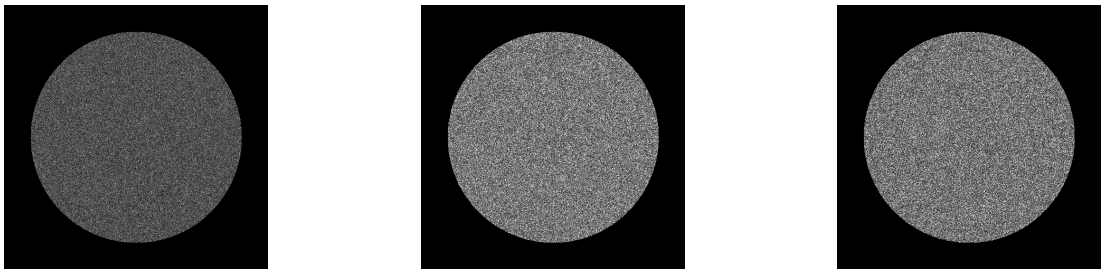
The normal of the joint plane is $(\mathbf{l}_3-\mathbf{l}_2)$ represented by $(\cos\eta_3-\cos\eta_2)$. The plane is visible in Figure 3.5(b) and the pattern of $|\mathbf{d}|$ is similar to the single-light tilt because of a similar direction of $(\mathbf{l}_3-\mathbf{l}_2)$.

In both Experiments E, F the zero-error plane does not move with the enlarging discrepancy $(v_{b3S}-1)$ because the discrepancy does not change the orientation of the plane normal. However, a change of $(v_{b3S}-1)$ influences $|\mathbf{d}|$ in a linear manner which is illustrated by RMS curves in Figure 3.5(c,d). Notice that a discrepancy in the intensity

of blue light leads to more severe errors than a discrepancy in the spectral relationship between blue sensor, green and blue light. RMS of $(a_S - a_R)$ and $\angle(\tilde{\mathbf{n}}_S, \tilde{\mathbf{n}}_R)$ show proportionally similar error for the albedo and the normal direction in the both experiments.

Image noise

The third set of experiments is focused on the image noise where the simulation-reconstruction chain follows Equation 3.19. The same matrices $\mathbf{V}, \mathbf{L}$ are used for the simulation and reconstruction of scaled normals. After simulating the captured image an RGB noise vector generated from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ is added to every pixel. In Experiment G an orthogonal light configuration is used instead of the configuration from previous experiments (the tilt angles ϕ_j are the same but the slant angle is $\theta = 54.73°$). This configuration is optimal with respect to image noise according to theoretical findings. The identity matrix $\mathbf{V}$ represents ideal light-sensor interaction. Figure 3.6(a) shows uniform noise in $|\mathbf{d}|$ across the hemisphere which proves no dependency of the error on a normal direction. The same observation can be made for Experiments H and I as well. RMS of $|\mathbf{d}|$ has a linear relationship with the standard deviation of noise σ (Figure 3.7(a)). The graph also confirms theoretical formulation of this relationship $RMS(|\mathbf{d}|)^2 = 3\sigma^2$ presented in [32].

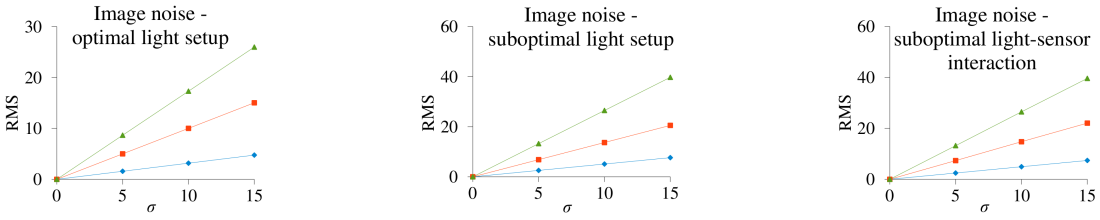


(a) (b) (c)

Figure 3.6: The magnitude of error in scaled normal $|\mathbf{d}|$ across the hemisphere for $\sigma = 10$: Exp. G - optimal light setup (a), Exp. H - suboptimal light setup (b) and Exp. I - suboptimal light-sensor interaction (c).

The original light configuration which has suboptimal light directions and the identity $\mathbf{V}$ are used in Experiment H. Figure 3.6(b) demonstrates amplified noise in comparison to

the optimal configuration. The fact that a noise vector Δ is enlarged by non-orthogonal $\mathbf{L}^{-1}$ is also proven by the increased RMS of $|\mathbf{d}|$ in Figure 3.7(b). Experiment I models suboptimal light-sensor interaction by setting the coefficients $v_{b2S} = v_{b3S} = 0.5$ in $\mathbf{V}$ (sensitivity of blue sensor spans spectra of green and blue light as in Experiment F). The matrix $\mathbf{L}$ is set according to the optimal light setup to examine solely an influence of $\mathbf{V}$. Similarly to Experiment H, the errors in scaled normals are higher than in the case of ideal matching between sensors and lights (Experiment G). It can be seen for all experiments that the error in albedo and normal direction also linearly depend on σ. The noise is propagated more to the albedo than to the normal direction.



(a) (b) (c)

Figure 3.7: The RMS graphs for different σ of image noise for Exp. G, H, I (a,b,c).

3.4.5 Discussion

The theoretical inference and the experiments on synthetic data showed that the error in scaled normal $\mathbf{d}$ varies depending on surface orientation in the presence of an inaccuracy in the illumination matrix $\mathbf{L}$ and interaction matrix $\mathbf{V}$. Complex behaviour of $\mathbf{d}$ across the hemisphere depends on the magnitude and direction of the discrepancy present. This indicates that the uncertainty of the scaled normal should vary with respect to its orientation given the uncertainty in the capture setup parameters. However, the direction of $\mathbf{d}$ changes smoothly over the hemisphere which suggests a low-frequency bias in the reconstructed normal and albedo map. This bias affects the overall shape of the surface represented by the normal map but it does not threaten the reconstruction of small geometrical detail. Only large calibration errors could cause significant flattening of the detail in some ranges of normal orientation. The magnitude of discrepancy vectors in $\mathbf{L}$ and $\mathbf{V}$ has a linear relationship with the overall RMS error of $|\mathbf{d}|$ for all

experiments. RMS errors related to **V** cannot be directly compared to the errors for **L** due to the different nature and magnitude of introduced discrepancies. Experiments E, F work with wider range of inaccuracy because it is more likely to make larger error in the estimation of **V** than **L** in practice. Image noise is propagated into scaled normals and their RMS error is linearly dependent on the standard deviation of noise. Unit normals are proportionally less affected than albedos when scaled normals are decomposed. This also holds for the majority of discrepancies in **V** and **L**. Since the main interest is in reconstructing correct shape rather than appearance, this observation has positive implications.

The magnitude and spatial distribution of **d** is significantly influenced by the form of **L** and **V**. In the presence of image noise, the orthogonality of vectors forming both matrices guarantees minimal amplification of noise according to theoretical proofs. The overall error in scaled normals is minimal under these conditions for discrepancies in **V** and **L** as well. Two important guidelines can be identified on constructing a capture setup for PSCL from the theory and experimental evaluation. The directions of lights should be perpendicular and their spectral characteristics should match sensitivity curves of corresponding camera sensors.

3.5 Evaluation

Evaluation of the photometric stereo is conducted for facial performance capture. The capture setup described in Section 7.2 consists of multiple HD cameras, red, green and blue light source. The light configuration is not completely orthogonal (slant angle $\sim 24°$, deviations from equidistant spacing 120° between the lights). This is because of spatial limitations and trade-off between the orthogonality and a size of shadows on the illuminated face. Therefore, the matrix **L** is sub-optimal in terms of image noise and calibration errors. The spectra of colour lights are well matched with spectral sensitivity of corresponding camera sensors which results in almost diagonal **V**. Thus, the image noise and calibration errors are not amplified through sensor-light-material interaction. The error analysis in Section 3.4.4 is conducted using a virtual setup similar to the real one. Thus, this provides useful information about the magnitude of potential error in

normal estimation given discrepancies in the calibrated setup parameters.

The calibration of **L** assumes distant, directional light sources. However, in practice the used light sources do not provide perfectly directional illumination as they are relatively close given the size of capture volume. The direction of incident lights can deviate on edges of the capture volume from **L** estimated in the centre. This does not pose a significant problem since small imprecision in **L** does not have noticeable effect on reconstructed normals. The matrix **V** is estimated independently for each camera because of different sensor characteristics and colour balancing. The calibration of **V** requires all colour lights to be in the same position. In practice, only colour filters are changed on one of the light sources while the actor stays still. This assumes that all sources have the same light spectra.



Figure 3.8: Example frames from normal map sequence by PSCL for the dataset Martin-makeup2 - colour-coded (top row) and rendered with grey diffuse material (bottom row). Colour coding of normal coordinates in WCS: x - red (left to right), y - green (bottom to top) and z - blue (far to near).

3.5.1 Reconstruction of facial performance

Experiments are focused on reconstruction of facial normals rather than albedos because the focus of this work is geometry of the face. A time-varying normal map sequence is reconstructed by applying PSCL estimation (Equation 3.8) to one of the image sequences from the dataset Martin-makeup2 (described in Appendix G). The captured actor is uniformly painted with diffuse white make-up (Figure 3.9(a)) to comply with the assumption of uniform chromaticity. Other benefits of the make-up over plain skin are higher signal-to-noise ratio (SNR), no sub-surface scattering of light and reduction of specular highlights. The latter two interfere with the assumed Lambertian model of the surface. Markers are painted on the face for tracking purposes (more explanation in Appendix A).



(a) (b) (c) (d)

Figure 3.9: Reconstruction of the face by PSCL in frame 100 from the dataset Martin-makeup2 - input image (a), albedo map (b), colour-coded normal map (c), normal map rendered with grey diffuse material (d).

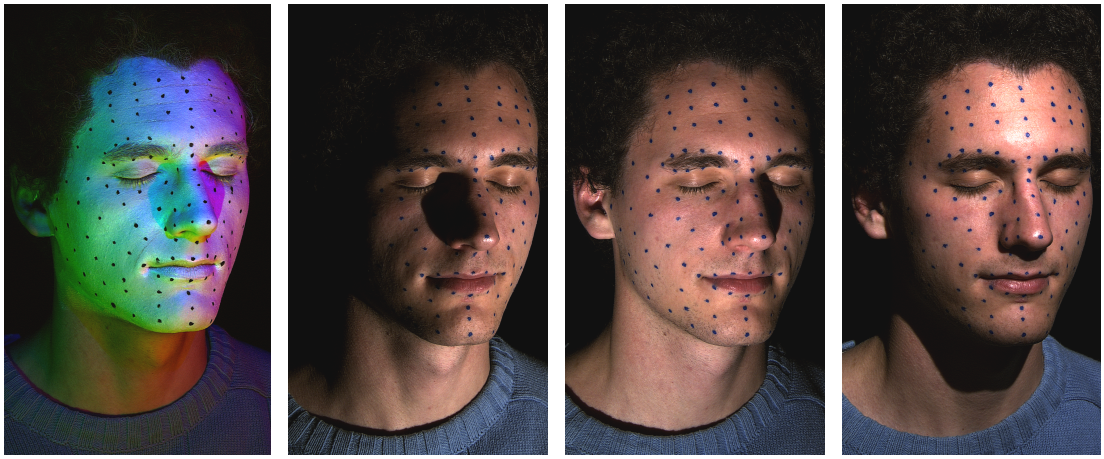
Figure 3.8 shows normal maps from one of the views in the dataset at example frames (the whole sequence in the supplementary video). Facial geometry is captured up to skin pore detail and fine wrinkles. The geometric detail is coherent over time and image noise is not very noticeable because of good SNR. The normal maps across the views are not completely aligned because of different low-frequency bias which is introduced by errors in view-dependent photometric calibration. The major artefacts

present in normal maps are caused by self-shadows (e.g. around nose, under chin). Missing constraints from occluded lights lead to incorrect calculation of normals in shadow regions shown by relighting of the face in Figure 3.8(bottom row). Incorrect normals are also noticeable on the eyes or inside of the mouth which are areas with non-Lambertian reflection of light. Lastly, markers are inconsistent with the rest of normal map because of dark appearance and different colour than the make-up. Figure 3.9 depicts PSCL result in more detail for a single frame. The albedo map in Figure 3.9(b) is fairly uniform across the face with some residual of local shading which indicates good separation between shape and appearance. However, the appearance of the face is not useful in practice due to the make-up covering the skin.

3.5.2 Comparison to photometric stereo with white lights

The method by Barsky and Petrou [8] working with colour images is used for PSWL (described in Section 3.1). The same three light sources without colour filters are used to secure comparable conditions with PSCL. Both methods use the same calibration data for $\mathbf{L}$ and assume the same intensity of light sources without colour filters (effectively $\mathbf{V}$ is the identity matrix for PSWL). A stationary actor with the neutral expression is captured under simultaneous colour illumination for PSCL (Figure 3.10 (a)) and time-multiplexed white illumination for PSWL (Figure 3.10 (b,c,d)). No make-up is applied on the face for PSWL because it does not assume uniform surface chromaticity.

Figure 3.11 compares the results from both techniques. The geometrical detail reconstructed by PSCL has similar quality to PSWL. Amount of noise in normals is similar in spite of lower intensity of colour illumination. This is due to the fact that white make-up preserves good SNR in the input image. PSWL suffers from shadow artefacts as well and additionally there are incorrect normals in regions with specular highlights (tip of nose and lips in Figure 3.11(bottom row)). The main disadvantage of PSWL is a risk of detail corruption because of actor's motion between measurements under individual lights. But real appearance of the face is obtained in colour albedo map.



(a) (b) (c) (d)

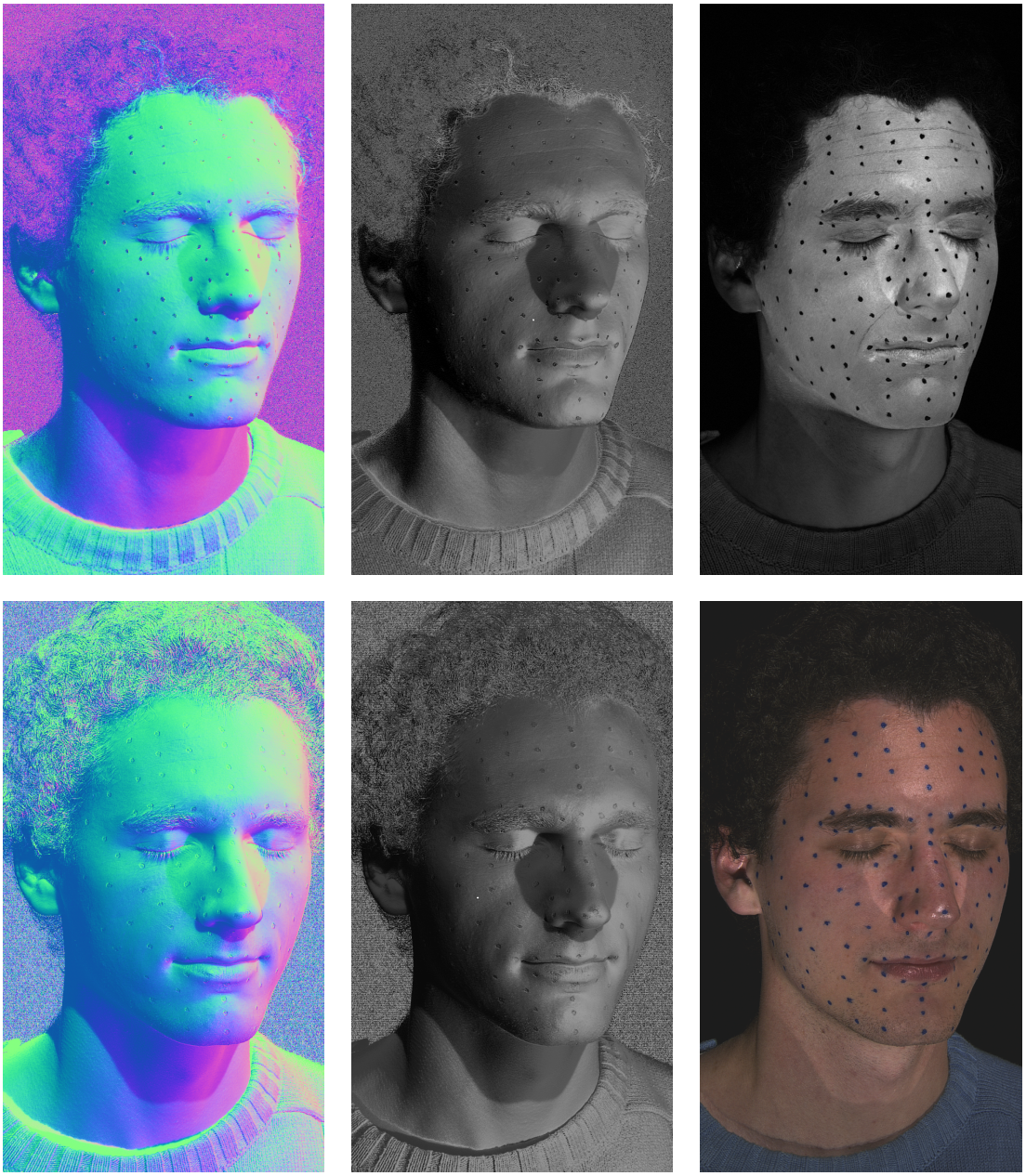
Figure 3.10: Input images - PSCL (a), PSWL (b-d).

3.5.3 Effect of facial make-up

To improve the performance of PSCL on fine skin details, white uniform make-up is applied on the face. PSCL on plain skin produces a significantly worse outcome than on white make-up as can be seen in Figure 3.12. This is also demonstrated by relighting with moving light source in the supplementary video. Several sources of imprecision prevent reconstruction of the finest level of skin detail. Firstly, the assumption of uniform chromaticity does not hold completely for human skin. Therefore, skin patches with chromaticities which deviate from the global mean given by the estimated $\mathbf{V}$ have different biases in the normal and albedo map. This can be perceived as coarse noise. Secondly, image noise has a larger impact because SNR is lower for the skin. Thirdly, normals are a bit blurred due to subsurface scattering of light in the skin. This affects mostly the red light, therefore the red channel of the image contains less high-frequency information. Lastly, specular reflections cause incorrect surface orientation (tip of the nose and the lips).

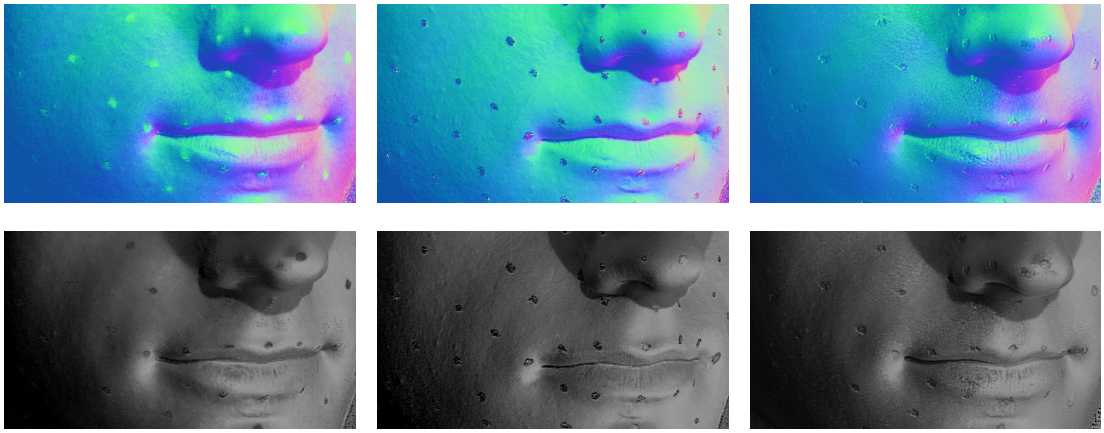
3.6 Conclusion

This chapter has investigated suitability of the photometric stereo with colour lights for geometry capture of dynamic facial detail. The geometrical detail is reconstructed up



(a) (b) (c)

Figure 3.11: Comparison of PSCL (top row) and PSWL (bottom row) - colour-coded normal map (a), normal map rendered with grey diffuse material (b), albedo map (c).



(a) (b) (c)

Figure 3.12: Comparison of PSCL without make-up (a), PSCL with make-up (b) and PSWL (c) - colour-coded normal map (top row), normal map rendered with grey diffuse material (bottom row).

to fine skin structure such as small wrinkles, pores. The normal maps are temporally coherent and obtained at the camera frame-rate. The quality of normals is comparable with standard photometric stereo with white lights if uniform make-up is applied on the actor's face. The calibration of light-sensor-material interaction has been improved over Hernandez et at. [48].

Error analysis of the photometric stereo with colour lights has been conducted which has not been previously addressed in the literature. In terms of image noise influence the optimality of an orthogonal illumination matrix is confirmed [32, 100, 102] and the same conclusion is drawn for a diagonal interaction matrix. Exact formulation of the relationship between calibration errors in the illumination and interaction matrices and the error in albedo-scaled normal is presented (in contrast to approximation through partial derivatives in [89, 58]). The form of this relationship suggests that orthogonality of vectors in the illumination and interaction matrices minimises the impact of discrepancies in their calibration. The theoretical conclusions with simulations of various errors provide valuable guidelines for design of a capture setup. They show that the calibration errors introduce only low-frequency bias in normals which does not hinder detail reconstruction. The normals are also proportionally less affected than albedo.

The main limitation of the presented technique is the assumption about uniform chromaticity of the surface and the resulting requirement of make-up on actor's face stemming from it. Another problem is low-frequency bias present in normals caused by photometric calibration errors which also differs across cameras. This prevents reconstruction of the correct facial shape by normal map integration. Furthermore, normal maps contain artefacts caused by shadows and very dark surface appearance. The low-frequency bias and shadow artefacts are addressed by a normal map correction using a reconstructed 3D mesh of the face which is presented in Section 7.5.

Photometric stereo provides a stream of normal maps in a common coordinate system from each camera at native frame-rate and resolution. This representation encodes all geometrical detail visible in original video streams, however it is still in the 2D domain. Therefore, stereo 3D reconstruction is employed to obtain a 3D mesh of the face with less detail at every frame. This mesh is combined with high-resolution normal maps to form high-detail facial 3D model as described in Chapter 7. The sequence of normal maps also lacks temporal consistency which is addressed in Chapters 4,5 and 6.

Chapter 4

Baseline sequential surface tracking

In the previous chapter the photometric stereo with colour lights is used to capture fine skin geometry at every frame but there is no temporal consistency across the data. Moreover, sequences of normal maps from multiple views do not form a full 3D model of the face. This chapter is focused on obtaining medium-scale facial geometry which has temporal consistency throughout the performance.

Temporal correspondence of 3D shape was initially addressed by scene flow methods [104, 65, 127, 86]. However, they rely on pre-computation of 2D optic flow in each view and provide only instantaneous flow field between a pair of frames. To acquire temporally consistent mesh sequences over longer periods of time, a template mesh can be deformed according to multi-view 2D optic flows and an unregistered geometry computed at every frame [122, 19, 47, 119]. A disadvantage is pre-computation of 3D geometries and optic flows at all frames. Also, independent optic flow computation in every view results in inconsistencies of 2D flows which decrease accuracy of the mesh deformation. *3D tracking* approaches [79, 23, 36] overcome these disadvantages by joint estimation of shape and motion directly in the 3D space. To achieve tractable optimisation, motion and shape of the surface are estimated using small 3D patches attached to the surface.

The best current 3D tracking method by Furukawa and Ponce [36] associates the surface patches with triangle fans around vertices of the mesh. Each patch has a fixed multi-view texture initialised at the reference frame. 3D tracking aligns each patch with images at the current frame using this texture. Independent optimisation of individual patches is aided by multi-view stereo initialisation and simple motion expansion across the surface. New locations of vertices are regularised by Laplacian smoothing combined with local mesh rigidity. The patch reference textures shrink and stretch together with the mesh deforming over time. This improves their sequential alignment with the images in all frames. The fixed texture limits drift of the mesh on the surface by referring back to the reference frame (track-to-first concept [21]). This method captures fairly complex motions of patterned surfaces and is able to recover from tracking errors and moderate occlusions. In the extension of this work for faces [37], a rigidity term in the regularisation is relaxed to accommodate extensive stretching and shrinking of human skin. Good results were reported only for faces with dense pattern make-up.

This chapter describes a baseline surface tracking method inspired by the work of Furukawa et al. [36, 37]. Building of a surface patch model for a template mesh at the reference frame is explained. 3D matching of a surface patch to another frame in multiple views is formulated and characteristics of the defined matching error are analysed. Optimisation of 3D patch matching providing surface motion estimates is described. The motion estimates constrain a weighted Laplacian deformation of the template mesh between two frames. Sequential tracking of the mesh chaining the frame-to-frame alignments yields a temporally consistent mesh sequence. The method is evaluated on facial performances under various conditions to identify limitations of the state-of-the-art in the surface tracking.

4.1 Problem formulation

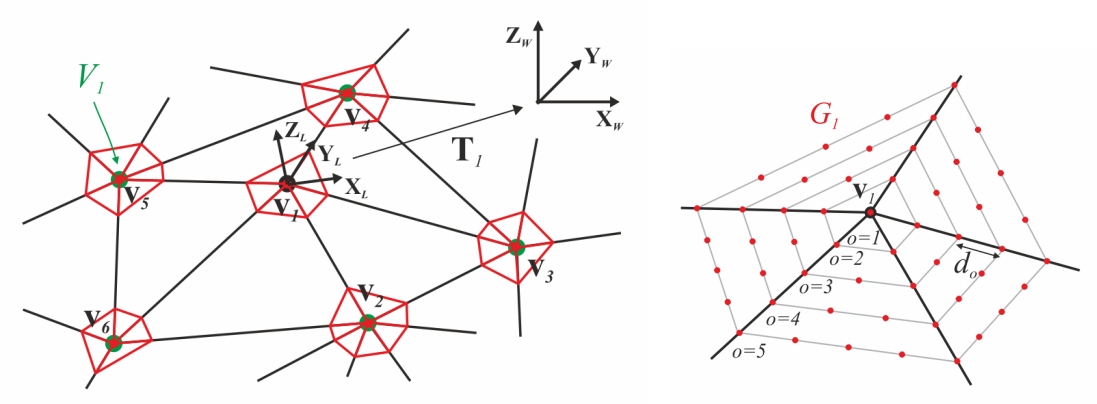
The input is a sequence of observations $\{O_t\}_{t=1}^{T}$ of a deforming surface for frames $\{1, ..., T\}$ where T is a number of frames in the sequence. Each observation O_t for a frame t consists of a set of images $\{I_t^c\}_{c=1}^{C}$. They are captured from multiple viewpoints by synchronised and calibrated cameras where C is a number of cameras. It is assumed

that each part of the surface to be reconstructed is observed by at least two cameras in time. The output is a mesh sequence $\{M_t\}_{t=1}^{T}$ where a triangular mesh M_t represents the surface at the frame t. The mesh $M_t = (X_t, \Gamma)$ consists of a set of vertex 3D positions X_t and a set of undirected edges Γ. The vertex positions $X_t = \{\mathbf{v}_i | \forall i \in [1..N]\}$ are defined in the world coordinate system (WCS) set by camera calibration (N is the number of vertices). The mesh topology $\Gamma \subseteq \{(i,j) | \forall i,j \in [1..N]; i \neq j\}$ is defined as a subset of all possible undirected edges among the vertices.

The output sequence $\{M_t\}_{t=1}^{T}$ is *temporally consistent*, therefore the vertex positions X_t correspond to the same set of surface points at every frame t and the topology Γ is fixed throughout the sequence. Initial placement of vertices on the surface and the mesh topology are designed by a user for the reference frame r ($r = t = 1$). The sequence $\{M_t\}_{t=1}^{T}$ is obtained by *sequential tracking* which concatenates *frame-to-frame non-rigid alignments* between successive frames $t-1$ and t. The frame-to-frame alignment estimates correspondence between observations O_{t-1} and O_t.

4.2 Surface patch model

A model of the observed surface is constructed at the reference frame r according to the concept presented by Furukawa and Ponce [36]. The surface is represented as a triangular mesh M where 1-neighbourhood of the vertex i is denoted $V_i = \{j | \forall (i,j) \in \Gamma\}$. Every vertex has a surface patch associated with it which is used to estimate motion of the vertex between frames (Figure 4.1). The pose of a patch in WCS needs to be represented independently from the mesh for estimation purposes. Therefore, each patch i has its own local coordinate system (LCS). The 4×4 transformation matrix $\mathbf{T}_i$ between LCS and WCS is formed from a translation vector $\mathbf{p}_i$ and a rotation vector $\mathbf{r}_i$ in axis-angle representation. The patch pose is initially defined in such a way that the origin of LCS coincides with the respective vertex i ($\mathbf{p}_i = \mathbf{v}_i$). The orientation $\mathbf{r}_i$ of LCS is such that aligns $\mathbf{Z}_L$-axis with the vertex normal given by surrounding faces. $\mathbf{X}_L$ and $\mathbf{Y}_L$-axes are on the tangent plane such that $\mathbf{Y}_L = \mathbf{Z}_L \times \mathbf{X}_W, \mathbf{X}_L = \mathbf{Y}_L \times \mathbf{Z}_L$ where $\mathbf{X}_W$ is an axis of WCS. The pose $\mathbf{T}_i = (\mathbf{p}_i, \mathbf{r}_i)$ changes during motion estimation and the patch moves away from its corresponding vertex.



(a) (b)

Figure 4.1: Surface patches attached to every mesh vertex (a). Central vertex 1 has neighbours $V_1 = \{2, 3, 4, 5, 6\}$. Its patch is formed by sample grid G_1 (b) which lies on the triangle fan with V_1. The grid G_1 has size $O = 5$ and spacial spacing d_o. LCS of patch 1 (index L) is related to WCS (index W) through the transformation $\mathbf{T}_1$.

A patch i is formed by a vector of 3D points G_i (size $3 \times |G_i|$) which forms an irregular grid centered around the vertex i. The sample grid G_i is shaped according to immediate triangle fan given by V_i of the vertex i (example in Figure 4.1). The sample points lie on the adjacent faces along N_o rings with increasing radius from the central vertex. Corner samples on mesh edges are spaced with a fixed distance d_o from $\mathbf{p}_i$ in 3D space. A value of d_o is chosen such that rings of any patch do not project further than 1 pixel apart in every view ensuring the sampling of image information without aliasing. The number of sample points increases with ring index o (central sample has $o = 1$). The ring o contains $o-2$ uniformly spaced points between the corner samples on every face. Sample grids can extend beyond the boundaries of the adjacent triangle fans but it still follows the triangle planes. Point locations are stored in barycentric coordinates with respect to the triangles they lie in. Benefits of this representation are that the sample grid automatically changes shape with a modification of V_i and actual 3D positions in G_i can be easily recomputed. The 3D positions are expressed in LCS, so that a change of $\mathbf{p}_i$ and $\mathbf{r}_i$ leads to a movement of the entire sample grid in a rigid manner.

Each patch has a visibility set $Q_i = \{c|c \in [1..C]\}$ which initially contains view indices where the vertex i is not self-occluded by the mesh. The set is further restricted by the angle between the vertex normal ($\mathbf{Z}_L$-axis) and the flipped viewing direction of a camera which has to be lower than 70°. This avoids poor sampling of image information

when projecting the patch grid into side views. Each patch has a multi-view texture $\{B_i^c\}_{c=1}^C$ which models its appearance across views at the reference frame r. Each vector $B_i^c \in [0..255]^{|G_i|}$ contains grey-scale pixel values for the sample points G_i from the image I_r^c. The 3D points G_i are first converted to WCS using $\mathbf{T}_i$ and then they are projected into the view c to sample $B_i^c = S(I_r^c, \mathbf{T}_i G_i)$. The function $S()$ encapsulates the projection of 3D points $\mathbf{T}_i G_i$ onto image plane of the camera c using its calibration data and the sampling of an image I_r^c at the frame r. The obtained pixel values are bi-linearly interpolated at the projected sample points. The textures $\{B_i^c\}_{c=1}^C$ are valid only for the views $c \in Q_i$.

In the context of surface tracking the variables related to the vertices and patches change over time (such as $\mathbf{v}_i$). Thus, they are denoted as a function of time $\mathbf{v}_i(t)$ when required. For brevity of notation, note that $\mathbf{v}_i = \mathbf{v}_i(t)$ where t is the current frame.

4.3 Frame-to-frame non-rigid alignment

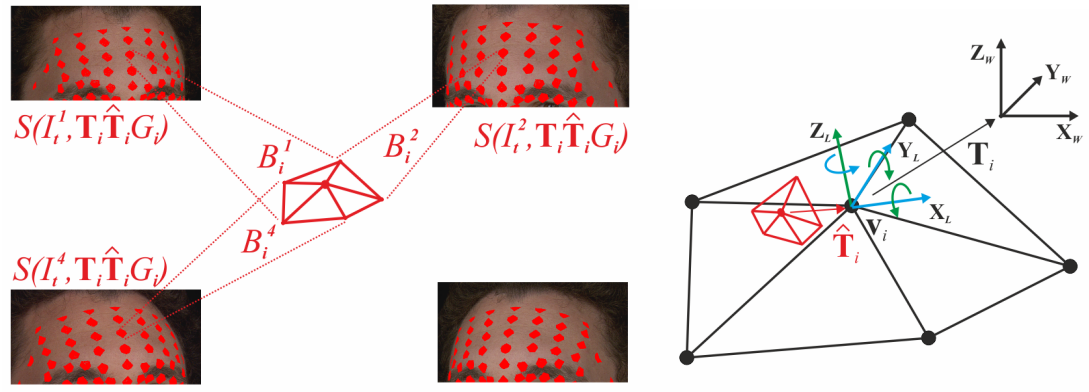
Alignment of surface observations O_{t-1} and O_t between successive frames is achieved in two stages. Firstly, displacements of individual mesh vertices are estimated by 3D matching of their respective patches to image information in O_t. Secondly, the resulting motion field for the whole mesh is regularised by global Laplacian deformation.

4.3.1 3D matching of surface patch

Finding a correspondence for the patch i between the frames $t-1$ and t is posed as a problem of aligning multi-view patch texture $\{B_i^c\}_{c=1}^C$ from the reference frame with images at the frame t. This problem is formulated as an optimisation task where the patch sample grid G_i is rigidly moved in 3D space from its initial pose $\mathbf{T}_i$ at frame $t-1$ to match its texture with the images $\{I_t^c\}_{c=1}^C$. Equation 4.1 defines an error function E_i for assessing multi-view alignment of $\{B_i^c\}_{c=1}^C$ with $\{I_t^c\}_{c=1}^C$ given a local modification of patch pose by $\hat{\mathbf{p}}_i, \hat{\mathbf{r}}_i$ (illustrated in Figure 4.2(a)).

$$E_i(\hat{\mathbf{p}}_i, \hat{\mathbf{r}}_i) \quad = \quad \frac{1}{|Q_i|} \sum_{c \in Q_i} \overline{NCC}(S(I_t^c, \mathbf{T}_i \hat{\mathbf{T}}_i G_i), B_i^c) \tag{4.1}$$

The 4×4 transformation matrix $\hat{\mathbf{T}}_i$ is formed from $\hat{\mathbf{p}}_i, \hat{\mathbf{r}}_i$ where a local translation $\hat{\mathbf{p}}_i$ shifts G_i from the origin of LCS along its axes. A rotation vector $\hat{\mathbf{r}}_i$ with Euler angles defines rotation of G_i around axes of LCS (order of rotations is $\mathbf{X}_L, \mathbf{Y}_L, \mathbf{Z}_L$-axis). The vectors $\mathbf{p}_i, \mathbf{r}_i$ (the matrix $\mathbf{T}_i$) are not directly optimised since they represent a global pose with respect to the WCS. To achieve a more meaningful movement of the patch grid reflecting local surface orientation, it is better to define a change of pose with respect to the patch LCS (Figure 4.2(b)). After the local modification $\hat{\mathbf{T}}_i$ the sample points G_i are expressed in the WCS through the transformation $\mathbf{T}_i$. The vectors of pixel values $S(I_t^c, \mathbf{T}_i\hat{\mathbf{T}}_i G_i)$ are obtained by projecting them to each view c in the visibility set Q_i. The grey-scale values sampled at the frame t are compared to the reference texture B_i^c using normalised cross-correlation (NCC). Note that $\overline{NCC} = 1 - (NCC + 1)/2$ is the inverted function which represents an error. The sum of matching errors across views is normalised by the size of the visibility set. The function $E_i(\hat{\mathbf{p}}_i, \hat{\mathbf{r}}_i)$ cannot be evaluated if $Q_i = \emptyset$ or $\mathbf{T}_i\hat{\mathbf{T}}_i G_i$ projects outside the image in any view from Q_i.



(a) (b)

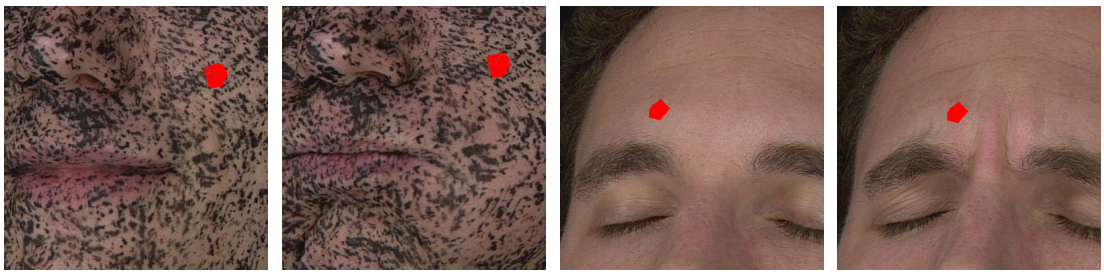
Figure 4.2: Multi-view alignment of patch textures $\{B_i^c\}_{c=1}^C$ with images $\{I_t^c\}_{c=1}^C$ where $Q_i = \{1, 2, 4\}$ (a). Global patch pose $\mathbf{T}_i$ with respect to WCS is modified by local transformation $\hat{\mathbf{T}}_i$ in LCS (b). Translation and rotation vectors $\hat{\mathbf{p}}_i, \hat{\mathbf{r}}_i$ forming $\hat{\mathbf{T}}_i$ can be decomposed into normal components (green) and tangential components (blue).

4.3.2 Analysis of matching error

The error function E_i (Equation 4.1) is analysed for an example patch to establish the complexity of its profile in various circumstances. The size of patch is $O = 20$ and

3D sampling distance $d_o = 0.2mm$. Sampling of parameters $\hat{\mathbf{p}}_i, \hat{\mathbf{r}}_i$ evaluates E_i for local deviations from initial patch pose $\mathbf{T}_i$. The function E_i is computed separately in the 3D space of local rotations $\hat{\mathbf{r}}_i$ and local translations $\hat{\mathbf{p}}_i$ for clarity of visualisation. In the first case, Euler angles in $\hat{\mathbf{r}}_i$ are sampled with the step $0.01rad$ in the range $\langle -1rad, 1rad \rangle$ and $\hat{\mathbf{p}}_i$ is fixed as a zero vector. In the second case, the components of $\hat{\mathbf{p}}_i$ are sampled with the step $0.1mm$ in the range $\langle -10mm, 10mm \rangle$ and $\hat{\mathbf{r}}_i$ is fixed as a zero vector. The function E_i is displayed across the 3D spaces of rotations and translations for the following experiments in the supplementary video. It can be assumed that the profile of E_i with $\hat{\mathbf{p}}_i, \hat{\mathbf{r}}_i$ varying simultaneously is more complex than in the separate cases.

The error profiles for local rotation and translation are computed in two situations. Firstly, the patch is matched at the reference frame r where its multi-view texture $\{B_i^c\}_{c=1}^C$ is sampled (using the initial pose $\mathbf{T}_i$). This is the ideal situation since the patch texture matches the images perfectly at $\mathbf{T}_i$. Secondly, the patch is matched using the reference texture at a frame t in which the surface has a different shape than at the frame r. The starting pose $\mathbf{T}_i$ is roughly estimated to place the patch over the same surface region as at the frame r. The estimate is not perfect to reflect a real scenario where the initial pose of patch is not completely correct before 3D matching at a particular frame.



(a) (b) (c) (d)

Figure 4.3: Example image from one of 4 views - the face with a random pattern at the reference frame r (a) and the frame t (b); the face with plain skin at the reference frame r (c) and the frame t (d). The patch used in experiments is visualised in red.

Face with a random pattern

Figures 4.3(a,b) show two frames from a facial performance of a subject painted with a random pattern which are used for the error analysis (the dataset Martin-pattern1). The error function E_i for the frame r is visualised in Figures 4.4(a-f). Figures 4.4(a-c) are slices of E_i for $\hat{\mathbf{r}}_i$ at the global minimum $[0\,0\,0]^T rad$ (the centre of slices). The value of the minimum is zero in this ideal case. The profile of E_i for $\hat{\mathbf{r}}_i$ is clear with the size of convergence basin approximately $-0.2 - 0.2rad$ $(-11.5° - 11.5°)$. The error E_i is more rugged for $\hat{\mathbf{p}}_i$ where the global minimum at $[0\ 0\ 0]^T mm$ is clear but quite localised. Its basin is around $-0.7 - 0.7mm$ wide and is surrounded by many local minima. This suggests that any gradient-based optimisation will require a good initialisation to converge to the global minimum. The error E_i for frame t is visualised in Figures 4.4(g-l). The global minima are located at $\hat{\mathbf{r}}_i = [-0.14\ 0.33\ -0.08]^T rad$ (value 0.226) and $\hat{\mathbf{p}}_i = [-0.2\ -0.5\ -1.0]^T mm$ (value 0.198). They are placed a bit off the centre since the initial $\mathbf{T}_i$ is not correct. The global minima in both spaces are weaker and have smaller basins around them in comparison to the frame r. This is caused by a change of surface appearance between the frames r and t due to a deformation. Thus, the patch texture does not perfectly match the information in individual images. If the surface deformation alters its appearance significantly, E_i becomes ambiguous with no clear global minimum.

Face with plain skin

To evaluate the influence of the amount of surface texture, the tests are performed on a facial performance without any make-up (the dataset Martin-skin1). Example images from the dataset are shown for the frames r and t in Figures 4.3(c,d). The function E_i for the reference frame r is visualised in Figures 4.5(a-f). The zero global minima for no rotation $\hat{\mathbf{r}}_i$ and no translation $\hat{\mathbf{p}}_i$ are easily identifiable. However, the extent of convergence basins is smaller than for the random pattern texture: the rotation around $-0.1 - 0.1rad$ $(-6.25° - 6.25°)$ and the translation around $-0.3 - 0.3mm$. Therefore, the initial pose of patch needs to be quite close to the global minimum for an optimisation algorithm to find it. The error E_i for the frame t is visualised in

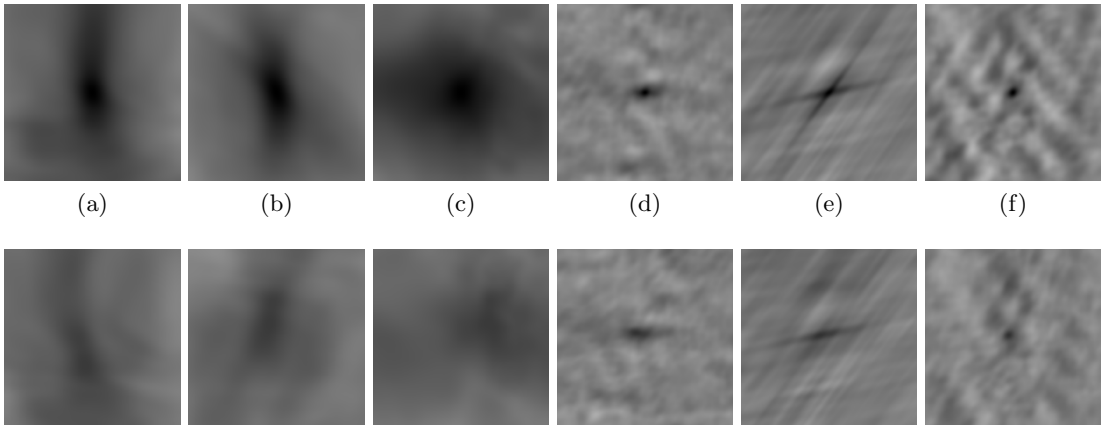


Figure 4.4: Slices through the error function E_i for the face with a random pattern at the reference frame r (top row) and at the frame t (bottom row). Local rotation of patch at r - slice with $\hat{\mathbf{r}}_i[x] = 0$ (a), $\hat{\mathbf{r}}_i[y] = 0$ (b) and $\hat{\mathbf{r}}_i[z] = 0$ (c). Local translation of patch at r - slice with $\hat{\mathbf{p}}_i[x] = 0$ (d), $\hat{\mathbf{p}}_i[y] = 0$ (e) and $\hat{\mathbf{p}}_i[z] = 0$ (f). Local rotation of patch at t - slice with $\hat{\mathbf{r}}_i[x] = -0.14$ (g), $\hat{\mathbf{r}}_i[y] = 0.33$ (h) and $\hat{\mathbf{r}}_i[z] = -0.08$ (i). Local translation of patch at t - slice with $\hat{\mathbf{p}}_i[x] = -0.2$ (j), $\hat{\mathbf{p}}_i[y] = -0.5$ (k) and $\hat{\mathbf{p}}_i[z] = -1$ (l). Range of error values $\langle 0, 1 \rangle$ is mapped into grey scale (0 - black, 1 - white).

Figures 4.5(g-l). The global minima are located at $\hat{\mathbf{r}}_i = [0.21\ -0.84\ -0.33]^T rad$ (value 0.228) and $\hat{\mathbf{p}}_i = [-4.8\ 1.3\ 3.7]^T mm$ (value 0.208). There is a large shallow basin across the range of $\hat{\mathbf{r}}_i$ but the location of the global minimum is not obvious. The error E_i with respect to $\hat{\mathbf{p}}_i$ has a complicated profile and contains multiple minima of similar value. Larger decrease in clarity of E_i between the frames r and t than for the random pattern is due to weaker skin texture which changes more dramatically with surface deformation (e.g. small wrinkles, skin folds, pore stretching). In general, weaker surface texture significantly increases ambiguity of the error function, so that matching using the fixed reference texture becomes a difficult optimisation task.

Variants of the matching error

The characteristics of E_i have also been investigated for different formulations of Equation 4.1. Multi-view patch texture $\{B_i^c\}_{c=1}^C$ can be replaced by a single texture B_i which is sampled at the reference frame r from a view with the least foreshortening of the projected sampling grid. Figure 4.6 depicts the altered E_i computed at the frame r for the face with a random pattern and plain skin. The global minima and their

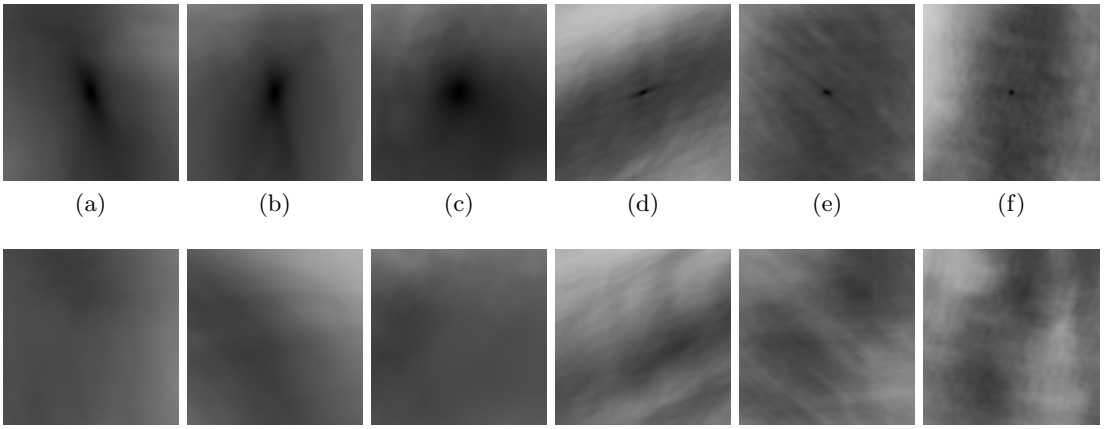


Figure 4.5: Slices through the error function E_i for the face with plain skin at the reference frame r (top row) and at the frame t (bottom row). Local rotation of patch at r - slice with $\hat{\mathbf{r}}_i[x] = 0$ (a), $\hat{\mathbf{r}}_i[y] = 0$ (b) and $\hat{\mathbf{r}}_i[z] = 0$ (c). Local translation of patch at r - slice with $\hat{\mathbf{p}}_i[x] = 0$ (d), $\hat{\mathbf{p}}_i[y] = 0$ (e) and $\hat{\mathbf{p}}_i[z] = 0$ (f). Local rotation of patch at t - slice with $\hat{\mathbf{r}}_i[x] = 0.21$ (g), $\hat{\mathbf{r}}_i[y] = -0.84$ (h) and $\hat{\mathbf{r}}_i[z] = -0.33$ (i). Local translation of patch at t - slice with $\hat{\mathbf{p}}_i[x] = -4.8$ (j), $\hat{\mathbf{p}}_i[y] = 1.3$ (k) and $\hat{\mathbf{p}}_i[z] = 3.7$ (l).

basins are generally weaker, especially for plain skin. The causes are different colour balance across cameras and different density of image sampling depending on patch pose with respect to individual cameras. Taking these into account, the correlation of single-view texture with the images is lower in comparison to the multi-view texture. The superiority of multi-view texture has also been observed in results of actual surface tracking.

Another possible alteration of calculating E_i changes the correlation measure. NCC in Equation 4.1 works with grey-scale pixel values, however full RGB information can be utilised. NCC can be computed in each colour channel separately and then averaged across them (note that this is 3 times more costly). Figure 4.7(a-f) shows the altered E_i at frame r for the face with plain skin. The function is very similar to NCC on grey-scale (Figure 4.5(a-f)) and no improvement is observed in the surface tracking. The use of colour information does not bring a significant benefit for skin under white illumination. NCC can also be replaced by computationally less expensive sum of squared differences (SSD). Figure 4.7(g-l) shows an example using grey-scale values. The error E_i for local rotation is similar to NCC (based on grey-scale or colour values).

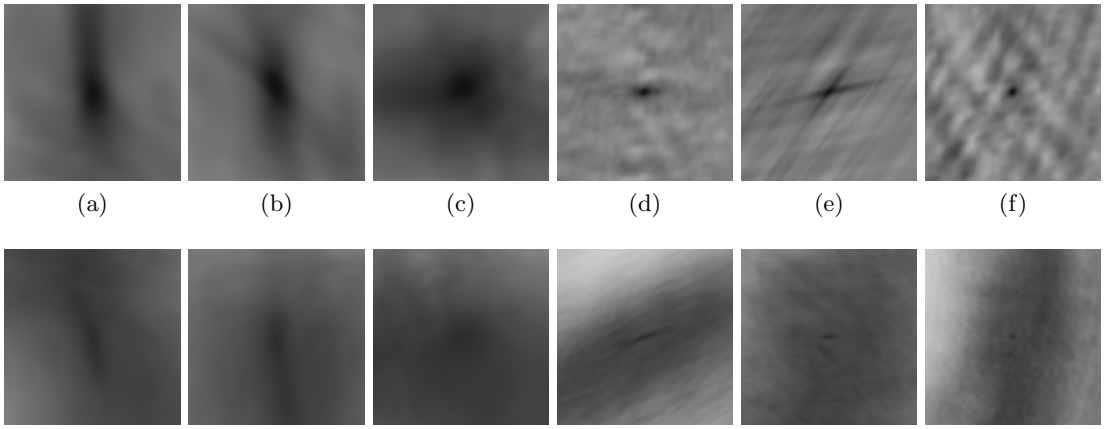


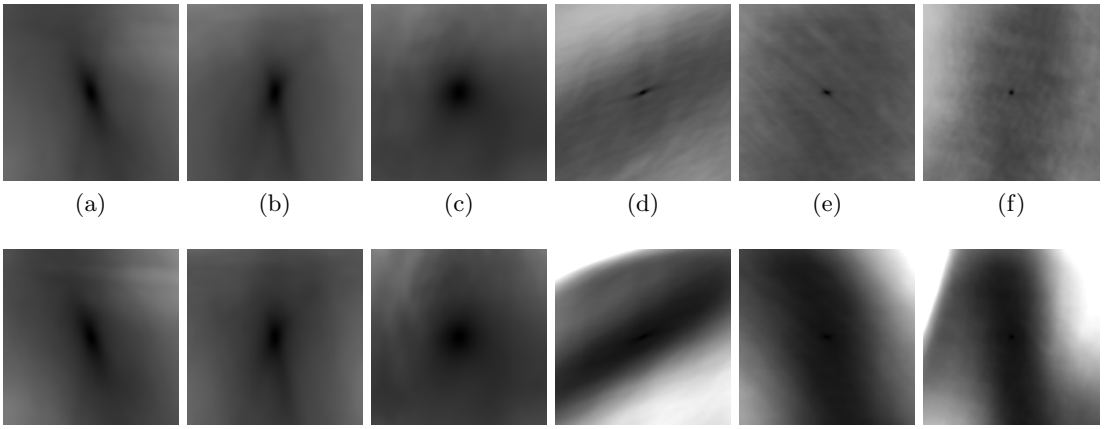
Figure 4.6: Slices through the error function E_i based on the single-view patch texture. Evaluated at the reference frame r for the face with a random pattern (top row) and for the face with plain skin (bottom row). Local rotation of patch at r - slice with $\hat{\mathbf{r}}_i[x] = 0$ (a,g), $\hat{\mathbf{r}}_i[y] = 0$ (b,h) and $\hat{\mathbf{r}}_i[z] = 0$ (c,i). Local translation of patch at r - slice with $\hat{\mathbf{p}}_i[x] = 0$ (d,j), $\hat{\mathbf{p}}_i[y] = 0$ (e,k) and $\hat{\mathbf{p}}_i[z] = 0$ (f,l).

The profile for local translation contains a broader convergence basin but the global minimum is less apparent. Larger difference for $\hat{\mathbf{p}}_i$ is present because the patch is shifted to surrounding surface regions which have different mean level of intensity. SSD is not robust to a linear transformation of the signal in contrast to NCC, therefore the error increases. A worse performance is observed with SSD in comparison to NCC for the surface tracking.

4.3.3 Optimisation of 3D patch matching

Error function E_i of each patch i is optimised by *independent gradient descent* (IGD). Initial pose $\mathbf{T}_i$ is given by the mesh M_{t-1} at the previous frame: $\mathbf{p}_i$ coincides with the position $\mathbf{v}_i(t-1)$ of the respective vertex at $t-1$ and $\mathbf{r}_i$ is set according to the shape of adjacent triangle fan in M_{t-1}. The local translation $\hat{\mathbf{p}}_i$ and rotation $\hat{\mathbf{r}}_i$ are optimised with respect to $\mathbf{T}_i$ in two stages following [36].

Firstly, only normal components of local modification are targeted: $\hat{\mathbf{p}}_i[z]$ - shift along the $\mathbf{Z}_L$-axis and $\hat{\mathbf{r}}_i[x], \hat{\mathbf{r}}_i[y]$ - Euler angles around $\mathbf{X}_L$ and $\mathbf{Y}_L$-axes (illustrated in green in Figure 4.2(b)). A change of normal components influences spatial position of the patch tangent plane with respect to the surface. This effectively changes the surface



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Figure 4.7: Slices through the error function E_i using NCC on colour (top row) and SSD on grey-scale (bottom row). Evaluated at the reference frame r for the face with plain skin. Local rotation of patch at r - slice with $\hat{\mathbf{r}}_i[x] = 0$ (a,g), $\hat{\mathbf{r}}_i[y] = 0$ (b,h) and $\hat{\mathbf{r}}_i[z] = 0$ (c,i). Local translation of patch at r - slice with $\hat{\mathbf{p}}_i[x] = 0$ (d,j), $\hat{\mathbf{p}}_i[y] = 0$ (e,k) and $\hat{\mathbf{p}}_i[z] = 0$ (f,l). SSD does not have a fixed range of values as NCC, thus the mapping to grey scale is set to get a good visualisation.

shape in contrast to the tangential components which slide the patch along the surface (illustrated in blue in Figure 4.2(b)). Thus, normal components are initialised by separate multi-view stereo optimisation in Equation 4.2 which refines only the shape.

$$E_i^S(\hat{\mathbf{p}}_i[z], \hat{\mathbf{r}}_i[x], \hat{\mathbf{r}}_i[y]) = \frac{1}{|Q_i|(|Q_i|-1)/2} \cdot \sum_{c \in Q_i} \sum_{c' \in Q_i, c' \neq c} \overline{NCC}(S(I_t^c, \mathbf{T}_i \hat{\mathbf{T}}_i G_i), S(I_t^{c'}, \mathbf{T}_i \hat{\mathbf{T}}_i G_i)) \tag{4.2}$$

The patch modified via $\hat{\mathbf{p}}_i[z], \hat{\mathbf{r}}_i[x], \hat{\mathbf{r}}_i[y]$ is projected into each pair of views c, c' from Q_i. The sampled pixels are compared by NCC across all possible image pairs $I_t^c, I_t^{c'}$. The sum of all matching errors is normalised by the total number of pairs. Note that the patch texture $\{B_i^c\}_{c=1}^C$ is not used at this stage. The optimisation of E_i^S provides more stable results than E_i limited to the normal components (Equation 4.1).

Secondly, the full vectors $\hat{\mathbf{p}}_i, \hat{\mathbf{r}}_i$ are simultaneously optimised based on the multi-view alignment of patch texture in Equation 4.1. The previous initialisation of normal components is meant to bring the starting point closer to a global minimum which can have a small basin of convergence as demonstrated in Section 4.3.2. Different

variants of optimised parameters have been tried for Equation 4.1. Optimising only position (either $\hat{\mathbf{p}}_i$ or $\mathbf{p}_i$) yields less stable motion estimates. Direct use of global pose parameters $\mathbf{p}_i, \mathbf{r}_i$ instead of local $\hat{\mathbf{p}}_i, \hat{\mathbf{r}}_i$ also leads to slightly worse outcome.

A multi-dimensional simplex method [78] is used for minimisation of the functions E_i^S and E_i. This algorithm is able to work with general non-convex functions without requiring knowledge of their derivative. Thus, it can be easily applied to the matching error with a complex profile demonstrated in Section 4.3.2. The initial size of the simplex is given by the magnitude of a perturbation from the starting value of the optimised parameters ($1mm$ for $\hat{\mathbf{p}}_i$, $0.1rad$ for $\hat{\mathbf{r}}_i$). Search for the best minimum continues until the size of the simplex (average distance between the centre of simplex and all its vertices) decreases under $0.01mm$ or the number of iterations exceeds a limit of 200. After convergence to a minimum the global patch pose $\mathbf{T}_i$ is updated by the final local modification as $\mathbf{T}_i \leftarrow \mathbf{T}_i\hat{\mathbf{T}}_i$. The value of the minimum represents a matching error e_i for the resulting pose $\mathbf{p}_i, \mathbf{r}_i$ of patch i. There is no guarantee that the global minimum is always reached due to the complex profile of E_i with many local minima.

Each patch has its own error function which is optimised independently from other patches. However, neighbouring points on the surface tend to have similar motion vectors (especially true for the smooth, continuous surface of the face). Therefore, Furukawa et al. [36] proposed additional initialisation of motion estimate for a patch from its already tracked neighbours before any optimisation takes place. In our case, only the position of the patch is initialised since E_i is more ambiguous in positional components (Section 4.3.2). The position $\mathbf{p}_i$ is updated using the adjacent patches from V_i^τ where a patch j is included if the matching error e_j is below a threshold τ_e. Equation 4.3 shows that $\mathbf{p}_i$ is shifted from $\mathbf{v}_i(t-1)$ by a weighted average of accurate motion estimates from the neighbours.

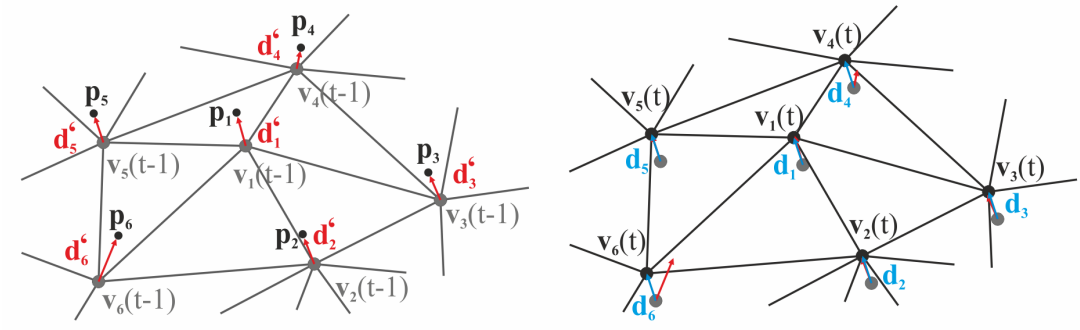
$$\begin{aligned} a_j &= \left(1 - \frac{\|\mathbf{v}_i(t-1) - \mathbf{v}_j(t-1)\|}{\sum_{j \in V_i^\tau} \|\mathbf{v}_i(t-1) - \mathbf{v}_j(t-1)\|}\right) \cdot \left(1 - \frac{e_j}{\sum_{j \in V_i^\tau} e_j}\right) \\ \mathbf{p}_i &= \mathbf{v}_i(t-1) + \frac{1}{\sum_{j \in V_i^\tau} a_j} \sum_{j \in V_i^\tau} a_j(\mathbf{p}_j - \mathbf{v}_j(t-1)) \end{aligned} \tag{4.3}$$

A weight a_j of each displacement is derived from the relative edge length in the previous

frame and relative matching error in the current frame. The weight a_j decreases with high error and large distance between patches. Patches with the largest number of already matched neighbours have a priority in the processing improves quality of the initialisation because a larger number of motion estimates is used.

4.3.4 Weighted Laplacian deformation

The 3D matching of patches produces a raw motion field described by 3D displacement vectors $\mathbf{d}'_i = \mathbf{p}_i - \mathbf{v}_i(t-1)$ as depicted in Figure 4.8(a). Note that the rotation vectors $\mathbf{r}_i$ are not used at this stage. The field can contain outliers due to inaccurate estimation of the current patch poses. The reasons can be ambiguity of the error functions (especially in regions with weak surface texture) and convergence into local minima. Also, the surface region represented by the patch can partially or completely disappear (e.g. eyelids, inner lips) which makes the correct matching impossible. A Laplacian deformation [99] is employed to regularise the raw motion field by filtering the outliers and ensuring spatial continuity of the motion. Laplacian mesh deformation tries to preserve the shape of the mesh M_{t-1} subject to the motion constraints $\mathbf{d}'_i$ weighted by their matching errors. The outcome is a new set of displacement vectors $\mathbf{d}_i = \mathbf{v}_i(t) - \mathbf{v}_i(t-1)$ which define the final $\mathbf{v}_i$ for the current frame t (Figure 4.8(b)).



(a) (b)

Figure 4.8: Motion estimates $\mathbf{d}'_i$ for vertices in M_{t-1} computed by 3D patch matching (a). Final displacements $\mathbf{d}_i$ defining the shape of M_t computed by Laplacian deformation (b). Note a significant difference of estimates $\mathbf{d}'_4, \mathbf{d}'_6$ from resulting displacements due to their down-weighting and inconsistency with overall surface motion.

The Laplacian deformation of the mesh is posed as a single optimisation problem across

all vertices in contrast to the per-patch 3D matching. Equation 4.4 formulates the functional which is minimised with respect to the displacements $\mathbf{d}_i$.

$$\underset{\mathbf{d}[x]}{\operatorname{argmin}} \quad s\|\tilde{\mathbf{L}}\mathbf{d}[x]\|^2 + \|\mathbf{W}(\mathbf{d}[x] - \mathbf{d}'[x])\|^2 \tag{4.4}$$

The problem is solved separately for each coordinate, thus $\mathbf{d}[x]$ denotes a vector containing x-coordinates of all $\mathbf{d}_i$ (similarly for y, z). The functional consists of smoothing and constraint terms which are weighted against each other by a smoothness coefficient s. The smoothing term regularises the motion field across the mesh using a discrete Laplacian operator. The $N \times N$ matrix $\mathbf{L}$ stacks up in rows the coefficients of Laplacian operators for individual vertices determined according to M_{t-1}. Multiplication $\mathbf{L}\mathbf{d}[x]$ then computes the Laplacian x-coordinate for each vertex [71]. The matrix $\tilde{\mathbf{L}}$ represents a linear combination of bi-Laplacian and Laplacian operator: $\tilde{\mathbf{L}} = ((1-\mu)\mathbf{L}^2 + \mu\mathbf{L})$. This is used to model bending (bi-Laplacian) and stretching (Laplacian) properties of the surface. The balance between them is set by a coefficient μ ($\mu = 0.6$ for skin according to the experiments). If the motion $\mathbf{d}[x]$ is smooth and preserves the shape of M_{t-1} (Laplacian coordinates do not change much), the magnitude of $\tilde{\mathbf{L}}\mathbf{d}[x]$ approaches zero.

The constraint term incorporates raw displacements $\mathbf{d}'_i$ in the form of soft constraints which have varying influence expressed by the diagonal weight matrix $\mathbf{W}$ of size $N \times N$. The weight w_i of a particular displacement $\mathbf{d}'_i$ is entered in the corresponding place on the diagonal of $\mathbf{W}$ as $\sqrt{w_i}$. The constraint weight is derived from the matching error e_i associated with $\mathbf{d}'_i$. The mapping between e_i and w_i described in Equation 4.5 is a declining linear ramp with a half-width δ_e centred around an error threshold ξ_e.

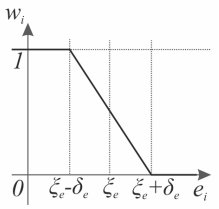


Figure 4.9: The mapping described in Equation 4.5.

$$w_i = \begin{cases} 0, & (e_i - \xi_e) > \delta_e \\ 1, & (e_i - \xi_e) < -\delta_e \\ -\frac{1}{2\delta_e}(e_i - \xi_e) + \frac{1}{2}, & |e_i - \xi_e| \leq \delta_e \end{cases} \tag{4.5}$$

In some situations, the error e_i can be low for outlying $\mathbf{d}'_i$, hence the estimate is not correctly down-weighted by w_i. This happens typically if the surface texture is repetitive and the patch is matched to the neighbouring region with very similar appearance. However, this kind of outliers occurs rarely according to our experiments and they do not have large magnitudes which would significantly influence overall deformation. The number of constraints can generally be smaller than the overall number of vertices due to failures in patch matching (e.g. full occlusion of a vertex). For the vertices without motion estimates the corresponding positions in $\mathbf{W}$ and $\mathbf{d}'[x]$ contain zero entries. Their final displacements are then derived purely from the motion of the whole mesh. If the final displacements $\mathbf{d}_i$ match closely the estimates $\mathbf{d}'_i$, the constraint term approaches zero.

Minimisation of both terms in Equation 4.4 is a least-squares problem which leads to an over-determined linear system in Equation 4.6.

$$\begin{bmatrix} \sqrt{s}\tilde{\mathbf{L}} \\ \mathbf{W} \end{bmatrix} \mathbf{d}[x] = \begin{bmatrix} \mathbf{0} \\ \mathbf{W}\mathbf{d}'[x] \end{bmatrix} \tag{4.6}$$

This system is solved by LU decomposition for each coordinate of the displacements $\mathbf{d}_i$ separately. The final vectors $\mathbf{d}_i$ represent the optimal motion field between frames given input constraints and assumed material properties of the surface. Iterative solving of displacements with local rotation update often included in the Laplacian deformation framework [98] is not used. The described scheme is sufficient since there are no significant rotations between successive instances of the facial surface.

The presented regularisation is simpler and can be solved more efficiently in contrast to Furukawa and Ponce [36]. The surface motion between frames is regularised instead of smoothing surface shape in the target frame t after raw motion estimation. Their approximate preservation of mesh shape from the reference frame prevents degradation of surface geometry but leads to non-linear optimisation. The non-linear optimisation is performed twice because the first pass with strong smoothness only identifies outlier constraints and the second pass regularises the mesh after filtering the outliers. The explicit weighting of constraints proposed in this work suppresses the outliers prior to the regularisation and thus retains a single-pass process.

4.4 Sequential tracking

The proposed frame-to-frame alignment is used to sequentially estimate motion of the surface starting from the reference frame r. The surface is represented by a mesh $M_r = (X_r, \Gamma)$ which shape and topology are defined by the user. The surface patch model is built using M_r and $\{I_r^c\}_{c=1}^C$ as described in Section 4.2. The mesh M_r needs to be accurate, so the patches are placed directly on the surface. Otherwise, multi-view textures $\{B_i^c\}_{c=1}^C$ would not be consistent with each other and represent different surface regions. The following steps are then repeated for each pair of successive frames $t-1$ and t starting from $r+1$.

1. The correspondences are estimated at frame t for all patches associated with M_{t-1} using the 3D patch matching based on IGD (Sections 4.3.1, 4.3.3).
2. The patch motion estimates drive the Laplacian deformation of M_{t-1} to M_t (Section 4.3.4).
3. The pose $\mathbf{T}_i$ is updated according to a new shape of M_t for every patch. $\mathbf{p}_i$ is set to a new value of $\mathbf{v}_i$ and $\mathbf{r}_i$ is changed to align the $\mathbf{Z}_L$-axis with a new normal at the vertex i and roughly preserve the direction of $\mathbf{X}_L, \mathbf{Y}_L$-axes in the frame $t-1$.
4. The visibility sets of patches Q_i are updated according to M_t.
5. The sample grids of patches are recomputed to reflect a shape change of the respective triangle fans. Note that new positions G_i are expressed in the updated LCS (the pose $\mathbf{T}_i$).

Every patch is treated as a rigid element during 3D matching in the frame t but its shape changes over time due to the step 5. This causes time-varying appearance of the reference patch textures when projected into views. The textures skew and rotate together with the patch motion; they shrink and expand together with the patch deformation. This improves the matching of the textures to the images in the case of non-rigid motion of the surface. The same accuracy cannot be achieved if the reference texture would be represented by a standard square window in the image plane. The 2D sample window does not have any robustness against changes of the surface pose and

shape in contrast to the 3D deformable sample grid. The temporally consistent mesh sequence $\{M_t\}_{t=1}^{T}$ with a fixed topology is a result of processing the whole sequence following the described algorithm.

4.5 Evaluation

Evaluation of the baseline surface tracking method is conducted for several facial performances captured under various conditions. The datasets differ in the nature of facial appearance: a fixed facial texture for Synthetic-skin1, a painted dense random pattern for Martin-pattern1, a set of painted markers for Martin-markers1 and plain skin for Martin-skin1. They have been recorded under uniform white illumination by the capture system described in Section 7.2. Technical description of the datasets can be found in Appendix G. All of them provide multi-view image sequences together with camera calibration data. The resolution of temporally consistent mesh computed for all performances is 2689 vertices and 5248 faces. Due to the nature of surface tracking results the reader is encouraged to assess them visually in the supplementary video. A common parameter configuration used for all datasets is: NCC on grey-scale,

	N_o	s	τ_e
Martin-pattern1	20	1.0	0.1
Martin-markers1	11	7.0	0.0
Martin-skin1	11	13.0	0.0
Synthetic-skin1	11	13.0	0.1

Table 4.1: Parameters for the baseline surface tracking across the evaluated datasets.

$d_o = 0.2mm$, $\xi_e = 0.1$, $\delta_e = 0.05$. Table 4.1 shows parameters varying across the evaluated datasets to reflect the amount of surface texture. The patch size N_o is lower for the plain skin than for the random pattern. It is more likely to match correctly smaller patches because the skin appearance due to surface deformations changes more than the pattern. The smoothness coefficient s increases for weaker surface texture to strengthen the regularisation which needs to deal with a larger portion of outliers in the estimated motion field. The initialisation step expanding low-error motion estimates has a positive impact only for the datasets Martin-pattern1 and Synthetic-skin1 where

the fixed patch texture works well (as reported in [36]). The other two datasets with predominantly natural skin texture on the face contain too many outlying displacements for a reliable motion expansion and the tracking becomes more unstable. The initialisation of 3D patch matching by separate multi-view stereo optimisation brings marginal improvement across the datasets according to further experiments.

4.5.1 Synthetic facial performance

Absence of ground-truth for real datasets is a common issue in the area of dense surface tracking. To allow quantitative evaluation, the dataset Synthetic-skin1 is artificially created. It is derived from a real performance to achieve realistic facial motion (the dataset Martin-skin2) . The temporally consistent mesh sequence is obtained by non-sequential tracking (Chapter 6) and temporally smoothed across cuts to improve the coherence over time. This sequence represents the ground-truth $\{M_t^{GT}\}_{t=1}^{T}$ which is textured with a fixed UV texture of the face from the reference frame. The fixed texture is not ideal since the real facial appearance changes over time but this avoids introduction of any inconsistencies between appearance changes and underlying motion of the geometry. The textured $\{M_t^{GT}\}_{t=1}^{T}$ is rendered into virtual views to obtain image sequences $\{\{I_t^c\}_{c=1}^{C}\}_{t=1}^{T}$.

The tracked mesh M_r is taken from $\{M_t^{GT}\}_{t=1}^{T}$ at the reference frame r so the resulting mesh sequence $\{M_t\}_{t=1}^{T}$ can be directly compared to the ground-truth on per-vertex basis. This dataset gives an advantage to the tested method because the surface texture does not change throughout the sequence which suits the track-to-first concept with fixed patch textures. Snapshots from the temporally consistent $\{M_t\}_{t=1}^{T}$ are presented in Figure 4.10. They demonstrate that the shape and motion are well recovered across the whole face. There is occasional drift of the mesh in the mouth region which is caused by severe distortion of the UV texture during mouth opening (noticeable at the frame 66). The accuracy of the result is illustrated by the ground-truth error in Figure 4.11. This is an average Euclidean distance between corresponding vertices of M_t and M_t^{GT}. The error fluctuates over time with peaks at the extremes of different expressions. The highest peak around frame 66 reflects the drift during the largest and

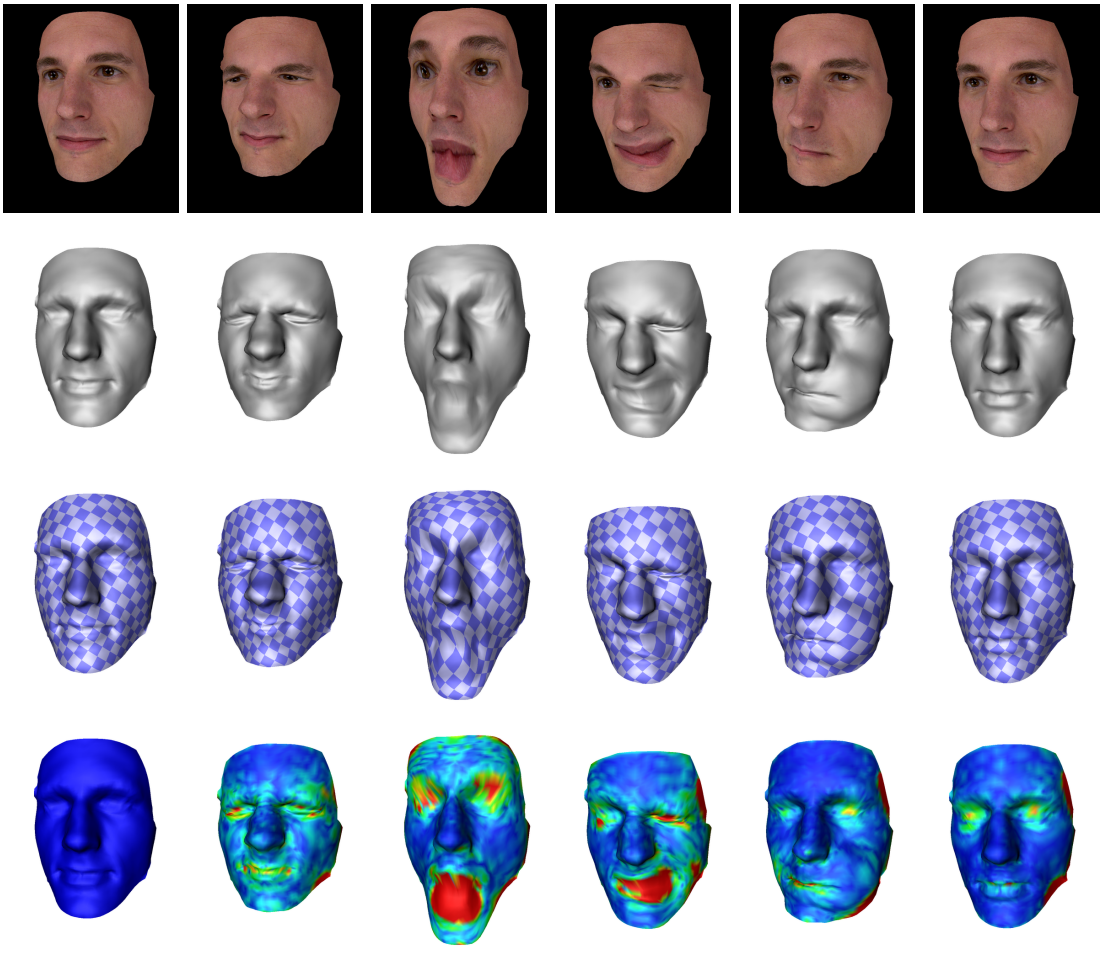


Figure 4.10: Snapshots from the temporally consistent mesh sequence for the dataset Synthetic-skin1: input images from one of the views (first row), meshes rendered with a uniform material (second row), meshes rendered with a fixed UV texture (third row) and difference to the ground truth (fourth row). Euclidean distance to corresponding vertices in the ground truth is visualised across the face (blue = $0mm$, red = $2mm$). The left most column represents the start/reference frame and the right most column the end frame of the sequence. Actual frame numbers are denoted.

most complicated deformation. The tracking algorithm recovers after each expression change, so there is no increasing trend which would suggest gradual accumulation of errors. The overall error across all vertices in every frame has a mean of $0.467mm$ and a standard deviation of $0.994mm$.

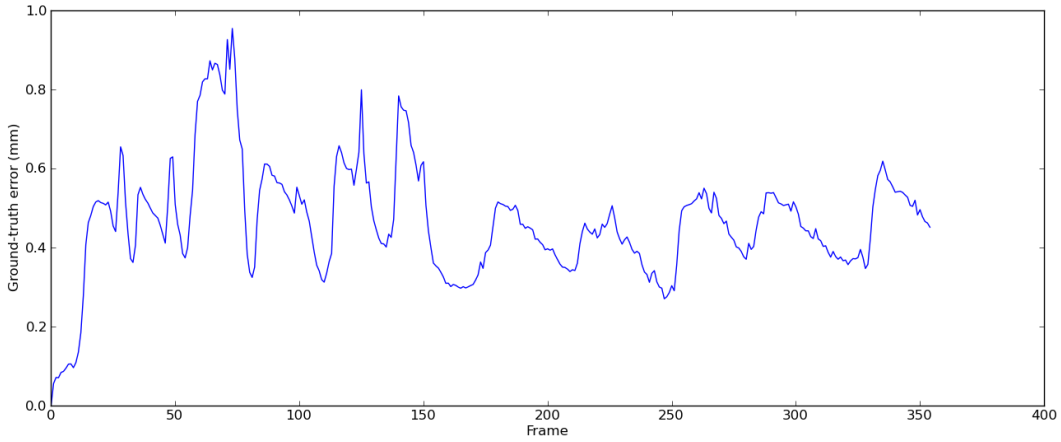


Figure 4.11: Average Euclidean distance across vertices to the ground-truth mesh sequence for the dataset Synthetic-skin1.

4.5.2 Influence of surface texture

The baseline surface tracking approach is evaluated in terms of the level of surface texture using the datasets Martin-pattern1, Martin-markers1 and Martin-skin1. The datasets contain a similar performance with comparable timing by a single actor. They differ in the amount of make-up painted on the actor's face: a dense random pattern for Martin-pattern1 (Figure 4.12), a set of markers for Martin-markers1 (Figure 4.13(top)) and plain skin for Martin-skin1 (Figure 4.13(bottom)).

Example frames from the temporally consistent mesh sequence for the dataset Martin-pattern1 in Figure 4.12 demonstrate correct capture of facial shape and its change over time. The shape details such as eye-brows wrinkling are recovered with temporal consistency (frame 182). The method is able to handle extensive surface deformations such as puffing out the cheeks or fast moving regions such as forehead and chin (frame 216). The only drawback are small shakes of the mesh in eye and mouth regions at the

extremes of some expressions. This is due to a significant change of appearance of the highly deformed regions in comparison to the patch reference textures.

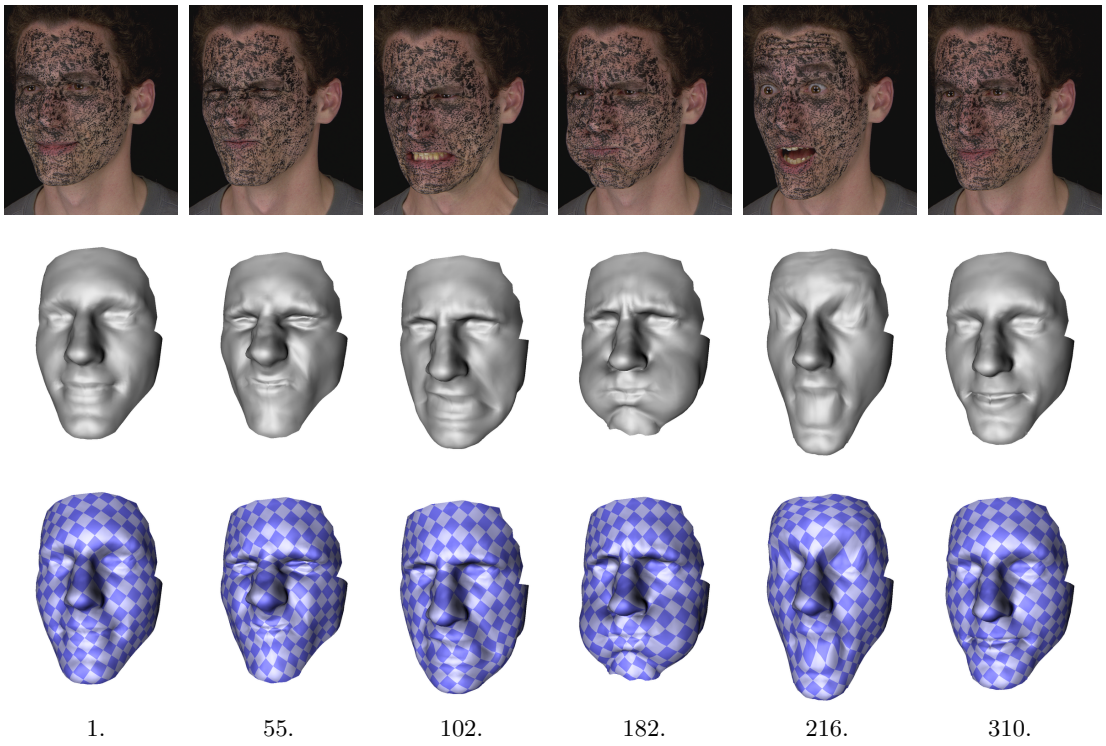


Figure 4.12: Snapshots from the temporally consistent mesh sequence for the dataset Martin-pattern1.

The dataset Martin-markers1 represents more difficult input and the quality of results decreases in comparison to Martin-pattern1 as shown in Figure 4.13(top). Mesh distortions appear during strong deformations and the method is not able to recover completely. Therefore, local drift increases throughout the sequence (notice the mouth and eyebrows at the last frame). The reason is that the appearance of plain skin differs considerably from the reference frame due to wrinkling, stretching etc. The increased amount of incorrect motion estimates needs to be handled by stronger regularisation which results in a smoother shape (e.g. eyebrows in frame 291).

The dataset Martin-skin1 poses the most challenging case which the baseline technique does not cope well with (Figure 4.13(bottom)). During large and rapid deformations the facial shape is visibly incorrect (e.g. puffing out the cheeks around frame 204 or surprise around frame 234). The mesh gradually degrades due to large distortions at each

expression. This sequence gives a significantly worse result in comparison to Martin-markers1 which is caused by the limited number of strong natural features in comparison to markers which provide better motion estimates. The dataset Synthetic-skin1 also contains plain skin as Martin-skin1 and provides even more challenging performance in terms of speed and amount of surface deformation. Despite that the temporal consistency is considerably better than for Martin-skin1. This demonstrates than the considerable variation of skin appearance over time is the main limiting factor for the method.

4.6 Conclusion

This chapter has described a generic surface tracking method based on the work of Furukawa and Ponce [36]. The contribution to their method is a different regularisation of raw motion estimates coming from 3D patch matching. The Laplacian deformation with soft constraints weighted by the matching errors has a linear formulation, and therefore can be solved more efficiently. Different variants of the patch matching error have been analysed in terms of their complexity in the parameter space for varying amount of the surface texture. A missing feature from [36] is coarse-to-fine optimisation over an image pyramid which does not improve the quality of tracking according to our experiments.

The presented baseline method has been evaluated for dense motion capture of a facial performance which poses a number of challenges such as rapid motion, complex non-rigid deformations and weak skin texture. The concept of tracking patches in the 3D space using multiple views has proven to be suitable given sufficient surface texture. This approach alleviates some issues of methods working primarily in the 2D image domain such as robustness against head pose change [47, 119] and inconsistent 2D optic flow estimation for different views [122, 19]. However, good results are achieved only for a well-textured surface (a face painted with a random pattern) or a surface without time-varying appearance (a synthetic face with a fixed texture).

The main limitation of the technique is low stability on weakly textured surfaces such as the skin. The approach does not fully recover from large errors appearing in the

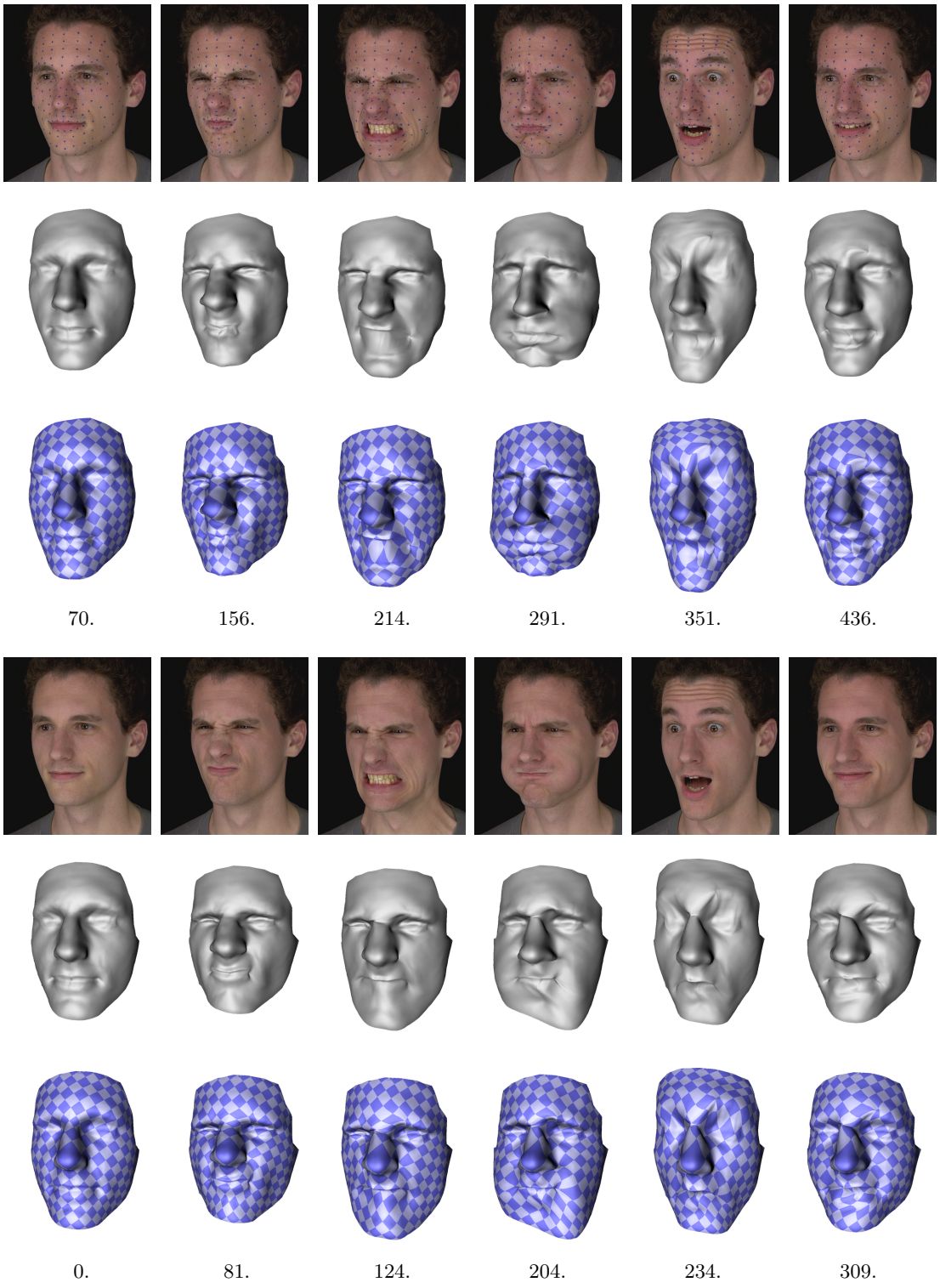


Figure 4.13: Snapshots from the temporally consistent mesh sequences for the datasets Martin-markers1 (top) and Martin-skin1 (bottom).

alignment during fast non-rigid motions which leads to drift of the tracked mesh. The key observation is that fine skin appearance varies extensively with changing expressions due to folds, wrinkling, pore stretching, etc. This cannot be modelled well by a deformable patch with a fixed reference texture even if it correctly changes the shape. The pattern make-up brings the needed constancy of appearance to some extent. But this also faces the same problem in highly deforming regions such as eyes and lips which manifests as occasional shakes in the mesh. Initialisation of the patch pose before the 3D matching does not help with this issue because the optimised objective function is already very ambiguous. The next chapter presents a robust sequential method which addresses these limitations and achieves superior performance on datasets with a weaker surface texture (markers, plain skin).

Chapter 5

Robust sequential surface tracking

In the previous chapter the baseline technique for surface tracking has been presented which is based on the work of Furukawa et al. [36]. Experimental evaluation identified limitations of this approach which prevent reliable tracking of the face with plain skin. Rapid complex motions of the face together with weak skin texture turn the motion estimation between frames into a difficult and ambiguous optimisation problem. Also, large variations of skin appearance occur due to expression change which cannot be modelled by a single reference texture. These factors lead to fast accumulation of alignment errors throughout a performance causing severe drift and degradation of the facial model.

Furukawa and Ponce [36, 37] advocate the track-to-first concept with the reference textures for patches to alleviate the drift problem. But they report accurate motion capture only for facial performances with an actor painted with a dense random pattern. The experiments in the previous chapter have demonstrated that this approach is not stable on the plain skin. Wilson et al. [119] aid tracking on plain skin with surface normals obtained by photometric stereo. Geometrical skin detail in the stream of normal maps is richer than skin texture in the original video which helps frame-to-frame optic flow computation. However, a noticeable drift still appears after concatenating flow fields over a number of frames (roughly 200 frames according to the authors).

Bradley et al. [19] correct the drift using additional optic flow estimation in the UV domain of the tracked mesh after an initial deformation. However, the results are not satisfactory in regions undergoing complex motion such as the mouth (even with a custom lip tracking algorithm).

To improve the baseline method, it is necessary to abandon the concept of a fixed appearance template and update the patch textures over time to reflect changes in the skin appearance. Image alignment of the textures from the previous frame makes the profile of the matching objective function less ambiguous (Section 5.2). But there is an increased risk that the patch textures adapt to a different 3D point on or even off the surface of the face as matching errors accumulate over time. The shape of the face reconstructed independently at every frame can limit motion of the patches onto the actual surface [122, 19]. However, drift along the surface needs to be addressed by an improved frame-to-frame patch matching. Cooperative optimisation of patch motion across the face proposed in this chapter brings significant improvement over independent gradient descent (IGD) used in the baseline technique. This optimisation is inspired by the PatchMatch algorithm from the image editing domain [6, 7].

The PatchMatch algorithm finds dense correspondences between two images. The correspondence of two pixels is defined by a similarity of square patches around them according to a chosen distance function. A displacement field with the resolution of the source image represents pairs of matching pixels which are the nearest neighbours in terms of similarity of their patches. This nearest neighbour field is computed by a fast approximate algorithm which iteratively updates the correspondences. The iterative matching has two phases: propagation which disseminates good solutions to adjacent pixels and random search which locally perturbs the current solutions. The convergence of the whole field depends on two key observations. Firstly, adjacent pixels usually have similar displacement vectors between the images. Secondly, large resolution of the field provides a high chance that a non-trivial number of pixels finds correct matches through the random sampling. Correct solutions are then spread across the field by the propagation. These assumptions are applicable to the problem of matching a dense set of 3D patches across a surface, thus the correspondence algorithm is relevant as well.

This chapter proposes a robust surface tracking method which improves over the baseline method in Chapter 4 for weakly textured surfaces such as skin. An extended objective function for 3D patch matching is formulated and its characteristics analysed for varying amount of the surface texture. Novel cooperative optimisation of patches across the face yields improved motion estimates for subsequent weighted Laplacian deformation. Coarse-to-fine scheme is proposed to increase robustness of the frame-to-frame alignment against large motions. Comprehensive evaluation of the robust method analyses the influence of surface texture and compares against to the baseline method. Furthermore, different variants of the method are evaluated to identify the importance of individual algorithmic features.

5.1 3D matching of surface patch

The objective to obtain a temporally consistent mesh sequence $\{M_t\}_{t=1}^{T}$ is the same as formulated in Section 4.1. However, the input observations O_t at each frame include the multi-view images $\{I_t^c\}_{c=1}^{C}$ and the mesh M_t^g which represents a shape estimate of the surface. The sequence $\{M_t^g\}_{t=1}^{T}$ is temporally unaligned, thus each mesh $M_t^g = (X_t^g, \Gamma_t^g)$ has time-varying vertex positions X_t^g and also time-varying topology Γ_t^g. The unaligned meshes can be reconstructed by an arbitrary multi-view stereo technique [92]. However, they should accurately model instantaneous shape of the surface because it constrains the surface tracking.

Finding correspondence for the patch i between the frames $t-1$ and t is reformulated as a two-fold problem in contrast to the baseline approach in Chapter 4. The alignment of multi-view patch texture $\{B_i^c\}_{c=1}^{C}$ with the images $\{I_t^c\}_{c=1}^{C}$ at frame t is combined with fitting the patch to the unaligned geometry M_t^g. The patch textures are updated at every frame. The texture adaptivity addresses inability of the fixed reference textures to cope with large changes of the surface appearance due to deformations (in the case of skin). However, the per-frame update of the textures increases the risk of drift since the patch can gradually adapt to a different 3D point as the tracking errors accumulate over time. To limit the drift, the motion of the patch between the frames $t-1$ and t is constrained to be in proximity to M_t^g similar to Zhang et al. [122].

Equation 5.1 defines a joint error function E_i^g which consists of image alignment (1.term) and geometry fitting (2.term) of the patch i.

$$E_i^g(\mathbf{p}_i) = \left(\frac{1}{|Q_i|} \sum_{c \in Q_i} \overline{NCC}(S(I_t^c, \mathbf{T}_i G_i), B_i^c(t-1)) \right) + w_g \rho(\|\mathbf{p}_i - \mathbf{g}_i\|, \sigma_g) \qquad (5.1)$$

The error E_i^g is minimised by altering directly the patch position $\mathbf{p}_i$ in comparison to Equation 4.1. An initial value of $\mathbf{p}_i$ coincides with a vertex position $\mathbf{v}_i(t-1)$ from M_{t-1}. The rotation vector $\mathbf{r}_i$ forming the transformation $\mathbf{T}_i$ together with $\mathbf{p}_i$ has a fixed value given by the shape of M_{t-1}. The vector $\mathbf{r}_i$ could be optimised together with $\mathbf{p}_i$ but there is only a marginal improvement of accuracy using the optimisation scheme explained in Section 5.3. The first term for alignment with the images $\{I_t^c\}_{c=1}^C$ is similar to Equation 4.1 except that $\{B_i^c\}_{c=1}^C$ comes from the previous frame $t-1$ rather than the reference frame r. Also, the global pose $\mathbf{T}_i$ with respect to WCS is not modified by a local transformation $\hat{\mathbf{T}}_i$ but it is altered directly through $\mathbf{p}_i$ instead.

The second term in Equation 5.1 forces the patch position $\mathbf{p}_i$ (associated with the central sample point) close to the unaligned mesh M_t^g as depicted in Figure 5.2. The point $\mathbf{g}_i$ is an approximation of the closest point to $\mathbf{p}_i$ on M_t^g. This can be efficiently computed using depth maps generated for M_t^g in each view. The view from the visibility set Q_i is selected according to the minimal angle between the patch normal and the flipped viewing direction of a camera. The position $\mathbf{p}_i$ is then projected into this view and $\mathbf{g}_i$ is determined by sampling the respective depth map at the point of projection. The distance between $\mathbf{p}_i$ and $\mathbf{g}_i$ is penalised by the Tukey bi-weight error norm ρ defined in Equation 5.2 [122].

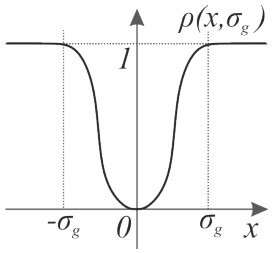


Figure 5.1: Tukey bi-weight error norm ρ defined in Equation 5.2.

$$\rho(x, \sigma_g) = \begin{cases} 3\frac{x^2}{\sigma_g^2} - 3\frac{x^4}{\sigma_g^4} + \frac{x^6}{\sigma_g^6}, & |x| \leq \sigma_g \\ 1, & |x| > \sigma_g \end{cases} \qquad (5.2)$$

The profile of ρ suggests that $\mathbf{p}_i$ is effectively restricted to a valley in the 3D space around the surface of M_t^g. The limited search space reduces the number of local minima complicating the optimisation of $\mathbf{p}_i$. The penalty is uniform beyond a distance σ_g which makes E_i^g less affected by large outliers in the raw geometry. The geometry fitting term and the image matching term are linearly combined by the weighting coefficient w_g which balances their influence. The function E_i^g cannot be evaluated for a particular value of $\mathbf{p}_i$ if $Q_i = \emptyset$, $\mathbf{T}_i G_i$ projects outside the image in any view from Q_i or $\mathbf{g}_i$ cannot be computed because of missing data in M_t^g.

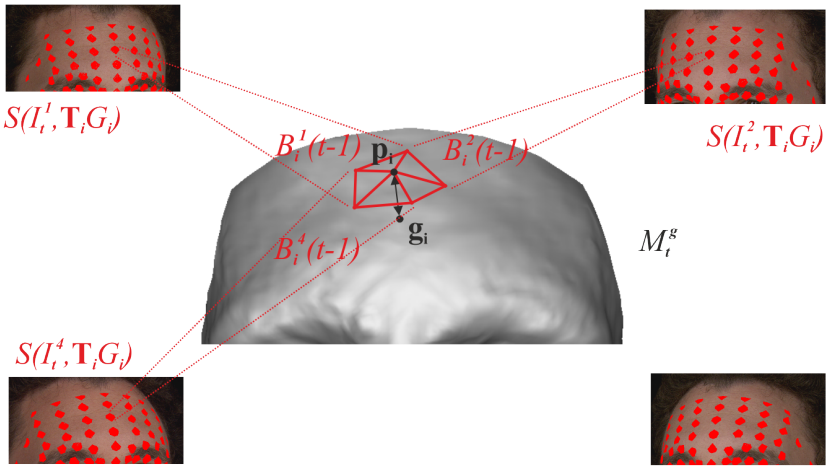


Figure 5.2: Multi-view alignment of patch textures $\{B_i^c\}_{c=1}^C$ from the previous frame $t-1$ with images $\{I_t^c\}_{c=1}^C$ and fitting the patch to an unregistered mesh M_t^g ($Q_i = \{1, 2, 4\}$).

5.2 Analysis of matching error

The error function E_i^g (Equation 5.1) is analysed for an example patch similarly as in Section 4.3.2. Again, the size of patch is $N_o = 20$ and 3D sampling distance $d_o = 0.2mm$. The raw mesh M_t^g necessary for geometry fitting is computed by the multi-view stereo method described in Section 7.3. The fitting parameters are set as follows: $\sigma_g = 10mm, w_g = 1.0$. Rotation of the patch is not evaluated since E_i^g optimises only the position $\mathbf{p}_i$. The vector $\mathbf{p}_i$ is not sampled directly for the error evaluation but the local translation $\hat{\mathbf{p}}_i$ is used to generate deviations from $\mathbf{p}_i$. This does not influence the

nature of the error function. The components of $\hat{\mathbf{p}}_i$ are sampled with a step $0.1mm$ in the range $\langle -10mm, 10mm \rangle$ and the global rotation $\mathbf{r}_i$ is fixed. The function E_i^g is displayed across the 3D space of translations for the following experiments in the supplementary video.

The error profiles for local translation are computed in two situations. Firstly, the patch is matched at the starting frame $t-1$ where its multi-view texture $\{B_i^c\}_{c=1}^C$ is also sampled (using the initial pose $\mathbf{T}_i$). This is the ideal situation since the patch texture matches the images perfectly at $\mathbf{T}_i$. Secondly, the patch is matched at the frame t using the texture from the frame $t-1$ which is a standard situation for tracking with the adaptive texture. The starting pose $\mathbf{T}_i$ is the same as at $t-1$ so the patch is naturally misaligned with the observations O_t.

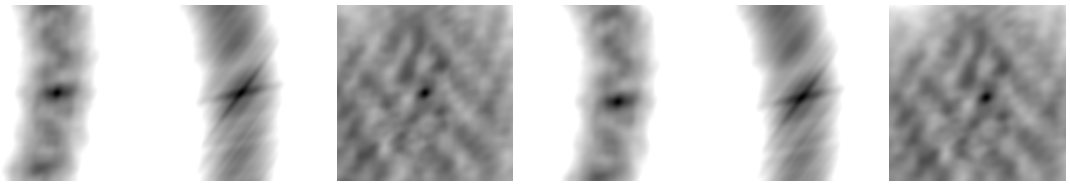


(a) (b) (c) (d)

Figure 5.3: Example image from one of 4 views - the face with a random pattern at the frame $t-1$ (a) and t (b); the face with a plain skin at the frame $t-1$ (c) and t (d). The patch used in experiments is visualised in red colour.

Figures 5.3(a,b) show two successive frames from a facial performance of the actor painted with a random pattern (the dataset Martin-pattern1). The frames selected for the error analysis have a fair amount of surface motion between them. The error E_i^g for the frame $t-1$ is visualised in Figures 5.4(a-c) which are slices through the function at the point of global minimum $[0\,0\,0]^T mm$ (zero value). The rugged profile with a small basin around the global minimum is similar to E_i from the baseline approach (Figures 4.4(d-f)) because the image alignment term is the same in this case. The difference is a clear valley across the examined volume given by the geometry fitting term. The valley in x, y-slices reflects the shape of M_i^g in this volume. The z-slice is along the tangent plane to the surface, thus the effect of the geometry term is not so visible. This shows

that the optimisation over E_i^g is constrained to the proximity of the true surface which suppresses many local minima present in the wider volume.

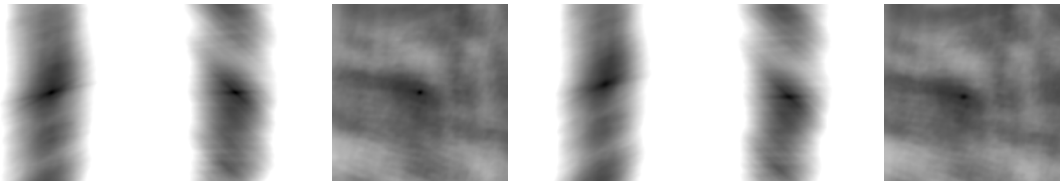


(a) (b) (c) (d) (e) (f)

Figure 5.4: Slices through the error function E_i^g for the face with a random pattern at the frame $t-1$ (a-c) and t (d-f). Local translation of patch at $t-1$ - slice with $\hat{\mathbf{p}}_i[x] = 0$ (a), $\hat{\mathbf{p}}_i[y] = 0$ (b) and $\hat{\mathbf{p}}_i[z] = 0$ (c). Local translation of patch at t - slice with $\hat{\mathbf{p}}_i[x] = 0.5$ (d), $\hat{\mathbf{p}}_i[y] = 1.0$ (e) and $\hat{\mathbf{p}}_i[z] = 1.0$ (f). Range of possible error values $\langle 0, 2\rangle$ is truncated at 1 and mapped into grey scale (0 - black, 1 - white).

The error E_i^g for the frame t is visualised in Figures 5.4(d-f). The global minimum is located at $\hat{\mathbf{p}}_i = [0.5\ 1.0\ 1.0]^T mm$ (value 0.014). The strength of the global minimum and overall quality of the function is comparable to the matching in the same frame $t-1$. This is due to small change of the surface appearance between successive frames. Therefore, E_i^g does not become as ambiguous as E_i where the reference texture can be matched to very different surface appearance at a distant frame (e.g. Figures 4.4(j-l)).

To make a comparison for plain skin, the tests are performed on a facial performance without any make-up (the dataset Martin-skin1). Example images from the dataset are shown for the frames $t-1$ and t in Figures 5.3(c,d). The error E_i^g for the frame $t-1$ visualised in Figures 5.5(a-c) has a very localised global minimum similar to E_i from the baseline approach (Figures 4.5(d-f)). The convergence basin is smaller than for the random pattern but it is also encompassed by the surface valley. The error E_i^g at the next frame t has a similar quality as at $t-1$ (Figures 5.5(d-f)). The global minimum located at $\hat{\mathbf{p}}_i = [0.5\ -0.9\ 0.01]^T mm$ is still quite clear with the low value 0.06. This is in contrast with the reference texture approach where the global minimum can often become unrecognisable (for example Figure 4.5(j-l)). Greater clarity of the error function with the adaptive patch textures over the fixed textures is important for achieving significantly more stable 3D patch matching on the plain skin than with the baseline technique.



(a) (b) (c) (d) (e) (f)

Figure 5.5: Slices through the error function E_i^g for the face with a plain skin at the frame $t-1$ (a-c) and t (d-f). Local translation of patch at $t-1$ - slice with $\hat{\mathbf{p}}_i[x] = 0$ (a), $\hat{\mathbf{p}}_i[y] = 0$ (b) and $\hat{\mathbf{p}}_i[z] = 0$ (c). Local translation of patch at t - slice with $\hat{\mathbf{p}}_i[x] = 0.5$ (d), $\hat{\mathbf{p}}_i[y] = -0.9$ (e) and $\hat{\mathbf{p}}_i[z] = 0.01$ (f).

5.3 Optimisation of 3D patch matching

The error function E_i^g is less ambiguous using adaptive patch textures as reported in Section 5.2. However, the convergence basin around the global minimum is still very small and is surrounded by many local minima, especially for the plain skin. Therefore, minimisation of E_i^g based on IGD (Section 4.3.3) is likely to reach a sub-optimal minimum. This results in lower stability of the surface tracking. To tackle this problem the PatchMatch correspondence algorithm [7] has been extended to the domain of surface tracking. The concept of matching image patches by *cooperative random sampling* (CRS) is adapted for 3D matching of surface patches. The essential assumption is that neighbouring patches on a surface move in a similar way. Thus, these patches can share intermediate solutions of their local minimisations performed by random sampling. This often prevents a descent into sub-optimal minima and improves the outcome of the local search. Generally, there is an increased likelihood that the patches across the whole surface converge to their individual global minima.

The initial pose $\mathbf{T}_i$ of a patch i is given by the mesh M_{t-1} at the previous frame. The position $\mathbf{p}_i$ with respect to WCS is directly optimised according to the error function E_i^g (Equation 5.1). The rotation $\mathbf{r}_i$ is fixed because E_i^g has a clear enough global minimum in the 3D space. Moreover, the risk of obtaining a sub-optimal solution increases with additional 3 rotational degrees of freedom. This has been experimentally observed especially during fast motions when patch error functions become more ambiguous. The optimisation of positions for all patches across the mesh is performed iteratively and in cooperation. At first, initial values of matching errors e_i are calculated at the

vertex positions $\mathbf{v}_i(t-1)$ using the images $\{I_t^c\}_{c=1}^C$. The patches are then processed one by one in H iterations and their positions $\mathbf{p}_i$ are modified from $\mathbf{v}_i(t-1)$ to decrease e_i. A solution for $\mathbf{p}_i$ is updated in two subsequent stages for every iteration.

Propagation stage: A patch i tries to adopt the current motion estimates from the adjacent patches in V_i which have already been processed in the current iteration (Figure 5.6(a)). A candidate for $\mathbf{p}_i$ is calculated by adding a displacement vector $\mathbf{p}_j - \mathbf{v}_j(t-1)$ from the neighbour j to the original position $\mathbf{v}_i(t-1)$. If the candidate has lower error on the function E_i^g than the current estimate $\mathbf{p}_i$, it is taken as a new solution and e_i is updated. Thus, the neighbouring patches are encouraged to have similar motion but this is not a hard constraint. An advantage of this approach is the ability to correctly recover motion discontinuities between different surface regions (such as lips, eye lids).

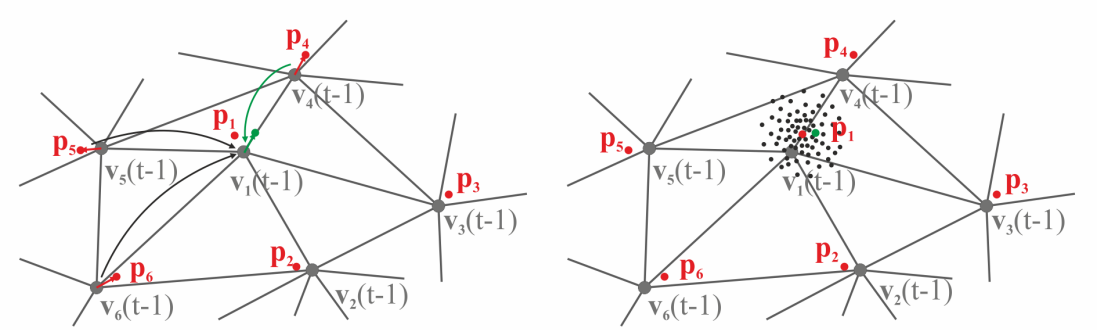


Figure 5.6: Cooperative random sampling - propagation stage (a), random sampling stage (b). In the propagation stage, the patch 1 tests motion estimates from the neighbours already processed in the current iteration (patches $4, 5, 6$). Its position $\mathbf{p}_1$ is updated according to the displacement from the patch 4 which gives a lower error than e_i. In the random sampling stage, a set of sample positions is generated for patch 1 around the current $\mathbf{p}_1$. Green sample has the lowest error below e_i, therefore it is selected as a new solution.

Random sampling stage: A local minimisation for a patch i is performed in the area around the current solution $\mathbf{p}_i$. New candidate positions are generated by random perturbation of $\mathbf{p}_i$: $\mathbf{p}_i + q_{max}\alpha^a\mathbf{u}$. $\mathbf{u}$ is a random 3D vector sampled from a uniform distribution in the interval $\langle -1, 1\rangle$ which is scaled by the current search range size. The size of range exponentially decreases by ratio $\alpha \in \langle 0, 1\rangle$ ($\alpha = 0.5$ in our experiments)

with increasing integer exponent a. For each value of a a fixed number U of candidate vectors is generated ($U = 5$ in our case) which results in a cloud of samples with increasing density towards the centre at $\mathbf{p}_i$ (Figure 5.6(b)). The range of random sampling is limited by the maximal bound q_{max} ($1mm$) and the increase of a from 0 is stopped when $q_{max}\alpha^a < q_{min}$ ($q_{min} = 0.1mm$). The function E_i^g is evaluated for each candidate and the one with the lowest error is compared with the current value of e_i. If it is better, $\mathbf{p}_i$ and e_i are updated.

The change of $\mathbf{p}_i$ from $\mathbf{v}_i(t-1)$ by both stages throughout all iterations is limited by a bounding box around $\mathbf{v}_i(t-1)$ with a half size q_{lim} ($q_{lim} > q_{max}$). This explicitly avoids motion estimates with magnitudes beyond possible motion between two frames. The reason is to prevent large outliers which can occasionally occur due to gradual traversal on E_i^g in wrong direction throughout iterations. The matching of a patch is unsuccessful if E_i^g cannot be evaluated at the initial position $\mathbf{v}_i(t-1)$ and at any candidate position suggested during the minimisation. The order in which the patches are processed by the propagation and random sampling stage is given by two rules. The next patch is selected according to: 1. the highest number of already processed neighbours in the current iteration, 2. the most promising neighbour in terms of the current error e_i. This ordering increases the impact of the propagation stage.

Interleaving the propagation and random sampling stage allows a patch to optimise on its own error function incrementally. The result of each local search is challenged by the motion estimates from the adjacent patches which can lead to further improvement. The overall solution across the whole surface improves with the increasing number of samples generated in the local search (influenced mostly by the parameter U) and increasing number of iterations H. The number H has a bigger influence since it facilitates greater propagation of estimates. However, all patches usually converge close to their minima in a few iterations ($H = 5$). There is no guarantee that all of them will reach their global minima. The likelihood of optimal outcome across the whole mesh improves with the resolution. A larger number of patches are more likely to find their optimal solutions and this is propagated across the surface.

The proposed approach has improved ability to avoid the convergence to local minima

on E_i^g than IGD (the variant Robust-IGD in Section 5.5.4). This greatly increases robustness to rapid non-rigid motions on weakly textured surfaces such as skin (Section 5.5.2). However, iterative processing increases computational load depending on values of U and H. The motion expansion by Furukawa et al. [36] used with IGD (Equation 4.3) differs from the propagation stage. It passes only a single averaged estimate and it is performed once before the local optimisation. This does not facilitate enough propagation of estimates. CRS is also different from the other non-rigid surface tracking schemes [86, 26] where all mesh vertices are optimised simultaneously in a single minimisation task with a large number of variables. Moreover, a global regularisation term is included which enforces smooth motion fields. CRS represents a more efficient scheme with small minimisation tasks per vertex which loosely cooperate with each other. This allows discontinuities in the motion field by separating out the global regularisation across the whole surface to the Laplacian mesh deformation.

5.4 Coarse-to-fine sequential tracking

Patch matching using CRS on the improved error function E_i^g is combined with the weighted Laplacian deformation in the same way as in the baseline method in Chapter 4. The sequential tracking is performed according to the algorithm described in Section 4.4 with the difference of updating multi-view patch textures. As the last step of processing at each frame, $\{B_i^c\}_{c=1}^C$ are sampled from the images $\{I_t^c\}_{c=1}^C$ after the shape of patch grids G_i is updated. The texture is obtained only for $c \in Q_i$, otherwise it is marked as invalid and the view is not used in for the matching at the next frame.

The sequential tracking of the mesh can be performed in a coarse-to-fine fashion extending the described scheme. The alignment between successive frames is iteratively refined by repeating 3D patch matching and Laplacian deformation across different mesh resolutions. This can be performed with both the baseline and robust alignment algorithm.

A coarse mesh M_r' provided by a user for the initial frame r gives a basis for the hierarchical model with L levels of detail (LOD) [79]. The mesh hierarchy $\{M_r^l\}_{l=1}^L$ is created by uniform subdivision of the mesh on the previous LOD starting from

$M_r^1 = M_r'$ (Figure 5.7). After each subdivision the mesh vertices are conformed to M_r^g to refine the shape of a new mesh M_r^l (more details in Section 7.4). Due to uniform subdivision the meshes on individual LOD share vertices ($X_r^1 \subset X_r^2 \subset ... \subset X_r^L$) but their topology is different ($\Gamma^1 \neq \Gamma^2 \neq ... \neq \Gamma^L$). The number of LOD L depends on the desired density of the finest mesh $M_r^L = M_r$ which is tracked over time. Separate sets of surface patches are created for each M_r^l. Because the vertex i can generally be included in several M_r^l, multiple patches can be associated with it. They have different sample grids G_i^l which are influenced by different V_i^l depending on the topology of M_r^l on a particular LOD l as depicted in Figure 5.7. Therefore, multi-view textures $\{B_i^{cl}\}_{c=1}^C$ of the patches related to the vertex i vary across LOD as well. All patches on every LOD are initialised at the frame r as described in Section 4.2 with one exception. The visibility set Q_i is shared by the patches related to the vertex i and it is set according to the finest mesh M_r^L.

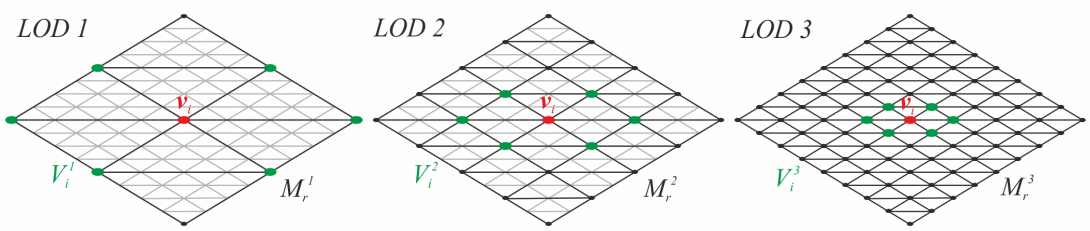


Figure 5.7: Hierarchical model with $L = 3$ LOD where a topology of M_r^1 is given by a user. The tracked full-resolution mesh M_r^3 is drawn in grey, the meshes M_r^l on individual LOD in black. Green set of vertices V_i^l shows the triangle fan around vertex $\mathbf{v}_i$ (red) which influences the patch associated with LOD l.

The tracking of M_t^L between the frames $t-1$ and t is performed incrementally by descending through the hierarchical model from the LOD 1 to L. On the LOD l the set of patches associated with M_t^l is matched between O_{t-1} and O_t and subsequently their displacements drive the Laplacian deformation of the full-resolution mesh M_t^L. Afterwards, the pose of patches associated with M_t^{l+1} is updated to align them with new shape of M_t^L. Progressing through LOD, the network of patches becomes denser as depicted in Figure 5.7 and the deformation of M_t^L is constrained by an increasing number of vertex displacements. On the LOD L every vertex has a patch providing a deformation constraint (as for the case without a hierarchical surface model). When

the final shape of M_t^L is obtained, the pose of all patches on every LOD can be updated accordingly. Their visibility sets are initialised for the next frame with respect to M_t^L as well. Finally, the sample grids of patches are recomputed according to respective M_t^l to reflect a change of the shape of related triangle fans.

During coarse-to-fine processing two key parameters vary: the patch size N_o and the smoothness coefficient s. Between two successive LOD N_o decreases by a factor ψ_o and s by factor ψ_s. The reason is that larger patches combined with stronger regularisation establish the correct overall motion of the surface on a coarser LOD (useful especially for large, fast deformations). Smaller patch size and weaker regularisation allows refining the surface motion on finer LOD.

The result is a temporally consistent mesh sequence $\{M_t\}_{t=1}^T$ where M_t corresponds to the finest LOD M_t^L of the mesh hierarchy. The coarse-to-fine processing provides marginal improvement when combined with the robust technique proposed in this chapter. The robustness of CRS is sufficient for motion estimation on the full-resolution mesh directly. There is a benefit in combining IGD with coarse-to-fine scheme as demonstrated in Section 5.5.5.

5.5 Evaluation

Evaluation of the robust surface tracking is conducted on the same set of facial performances as in Chapter 4 and an additional dataset Martin-skin2 which is more challenging terms of performance pace. Technical description of the datasets used (Martin-pattern1, Martin-markers1, Martin-skin1, Martin-skin2, Synthetic-skin1) can be found in Appendix G. All of them provide multi-view image sequences together with camera calibration data. Additionally, unaligned mesh sequences necessary for geometry fitting are pre-computed by the multi-view stereo method described in Section 7.3. The resolution of temporally consistent mesh computed for all datasets is 2689 vertices and 5248 faces.

Surface tracking results are standardly assessed visually in terms of temporal consistency (the reader is referred to the supplementary video). The reason is a lack of

quantitative measures for real data which would reliably evaluate drift in the obtained mesh sequences. In this work, two measures are tested for comparison of the results by different tracking methods - SAD error on unwrapped surface textures described in Section 5.5.3 and average Euclidean distance between equivalent frames in mirrored sequence described in Appendix B.

A common parameter configuration for all datasets is: NCC on grey-scale, $d_o = 0.2mm$, $N_o = 11$, $w_g = 1.0$, $\sigma_g = 10mm$. Note that coarse-to-fine processing is not used in the following experiments unless it is explicitly mentioned. Patch size N_o is fairly insensitive parameter for the robust method because larger textures sampled at the previous frame do not improve matching of a patch. Only small patch sizes ($N_o < 5$) cause more unstable results because of the lack of information content. Fitting to the unaligned mesh M_t^g has equal importance to the image alignment term by setting $w_g = 1.0$ (Equation 5.1). There is some loss of shape detail in the tracked mesh M_t in comparison to the raw M_t^g due to lower resolution and motion regularisation. Also the texture adaptation can gradually flatten larger wrinkles to some extent over a period of time. Stronger fitting ($w_g > 1.0$) can preserve more shape detail from M_t^g but it also causes more drift on the surface because of weakening the image matching term.

	s	ξ_e	δ_e	$q_{lim}(mm)$
Martin-pattern1	0.1	0.03	0.01	10
Martin-markers1	0.5	0.125	0.05	5
Martin-skin1	1.0	0.15	0.05	5
Martin-skin2	9.0	0.15	0.05	10
Synthetic-skin1	1.0	0.15	0.05	10

Table 5.1: Parameters for the robust surface tracking across the evaluated datasets.

Table 5.1 shows differences in certain parameters across the evaluated datasets depending on the surface texture. The smoothness coefficient s increases for weaker surface texture to strengthen the regularisation to deal with less accurate estimate of motion field. However, the regularisation is increased much less than for the baseline method (Table 4.1) because the patch matching is significantly better. The parameters ξ_e and δ_e for constraint weighting are lower for the datasets Martin-pattern1 and Martin-markers1 due to generally lower matching errors. The limit q_{lim} on magnitude of patch

displacements between frames is set to allow correct motion estimation of the fastest sections of the performance.

5.5.1 Synthetic facial performance

Quantitative evaluation is performed on the synthetic dataset Synthetic-skin1. Snapshots from the temporally consistent $\{M_t\}_{t=1}^{T}$ are presented in Figure 5.8. They demonstrate that the shape and motion are well recovered across the whole face. Local drift of the mesh in the mouth region is caused by severe distortion of the UV texture during mouth opening. The accuracy of results is illustrated by heat maps of the difference to the ground-truth mesh sequence in Figure 5.8(fourth row). Average ground-truth error across all vertices is plotted per frame in Figure 5.9. The overall error across all vertices in every frame has a mean of $0.740mm$ and a standard deviation of $0.981mm$.

The per-frame error fluctuates over time with peaks at the extremes of different expressions. The tracking does not fully recover after each expression change, thus there is an increasing trend which suggests accumulation of errors (visible in the heat maps in Figure 5.8 as well). This is caused by the adaptive patch textures which gradually adjust to different surface points, particularly during fast motions. The baseline technique achieves better overall accuracy because unchanging face texture in the dataset favours tracking with the fixed patch textures. However, error peaks for the baseline are relatively high due to substantial difference between the face appearance at extremes of emotions and the textures from the reference frame. This is visible as subtle shakes of the mesh in video at these points.

5.5.2 Influence of surface texture

The robust surface tracking is evaluated in terms of the level of surface texture on the datasets Martin-pattern1, Martin-markers1, Martin-skin1 and Martin-skin2 as the baseline approach in Section 4.5.2. Figure 5.10 demonstrates accurate capture of facial shape and its change over time for the dataset Martin-pattern1. Due to the strong texture, the result is comparable to the baseline surface tracking (Figure 4.12). However,

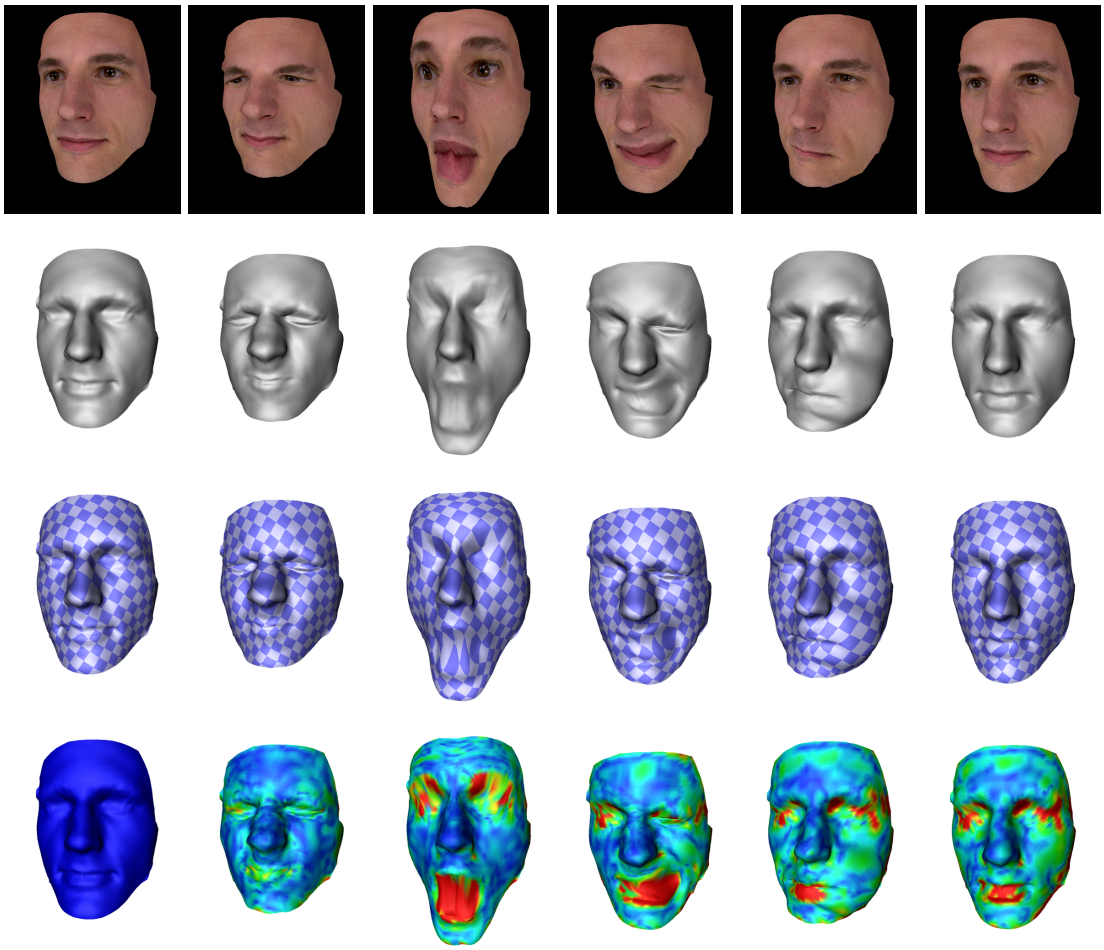


Figure 5.8: Snapshots from the temporally consistent mesh sequence for the dataset Synthetic-skin1: input images from one of the views (first row), meshes rendered with a uniform material (second row) and meshes rendered with a fixed UV texture attached at the reference frame (third row) and difference to the ground truth (fourth row). Euclidean distance to corresponding vertices in the ground truth is visualised across the face (blue $= 0mm$, red $= 2mm$). The left most column represents the start/reference frame and the right most column the end frame of the sequence. Actual frame numbers are denoted.

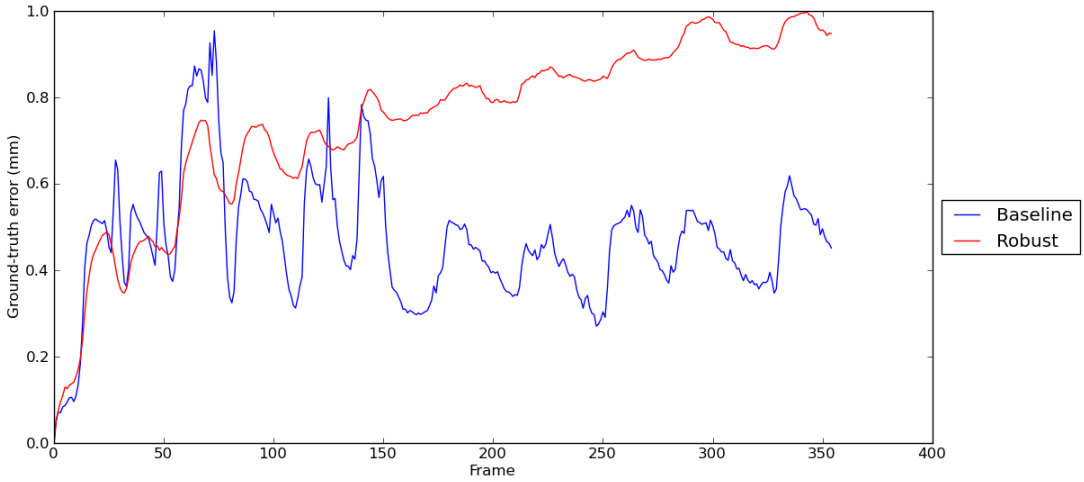


Figure 5.9: Average Euclidean distance across vertices to the ground truth for the dataset Synthetic-skin1.

small shakes observed during large deformations with the baseline method do not occur with the proposed robust method.

Despite weaker texture in the dataset Martin-markers1 the resulting mesh sequence (Figure 5.11(top)) has a similar quality to Martin-pattern1. The amount of shape detail is the same but there is a bit more local drift on the lips. CRS successfully propagates good matches from strong features such as markers to the skin areas between them. This leads to better temporal consistency and more precise facial shape than for the baseline method (Figure 4.13).

The dataset Martin-skin1 with plain skin is too challenging for the baseline approach (Figure 4.13). The proposed robust tracking is able to deal with weakness and large variation of skin texture over time and yields significantly better temporal alignment of the performance without large mesh distortions (Figure 5.11(bottom)). The reduced amount of strong features such as markers results in slightly smoother shape due to stronger regularisation. There is also a bit more mesh distortion in the mouth region than in the dataset Martin-markers1. The fast performance in the dataset Martin-skin2 (Figure 5.13(top) - variant Robust) shows limitations of the robust technique where noticeable drift is accumulated around the eyes and mouth.

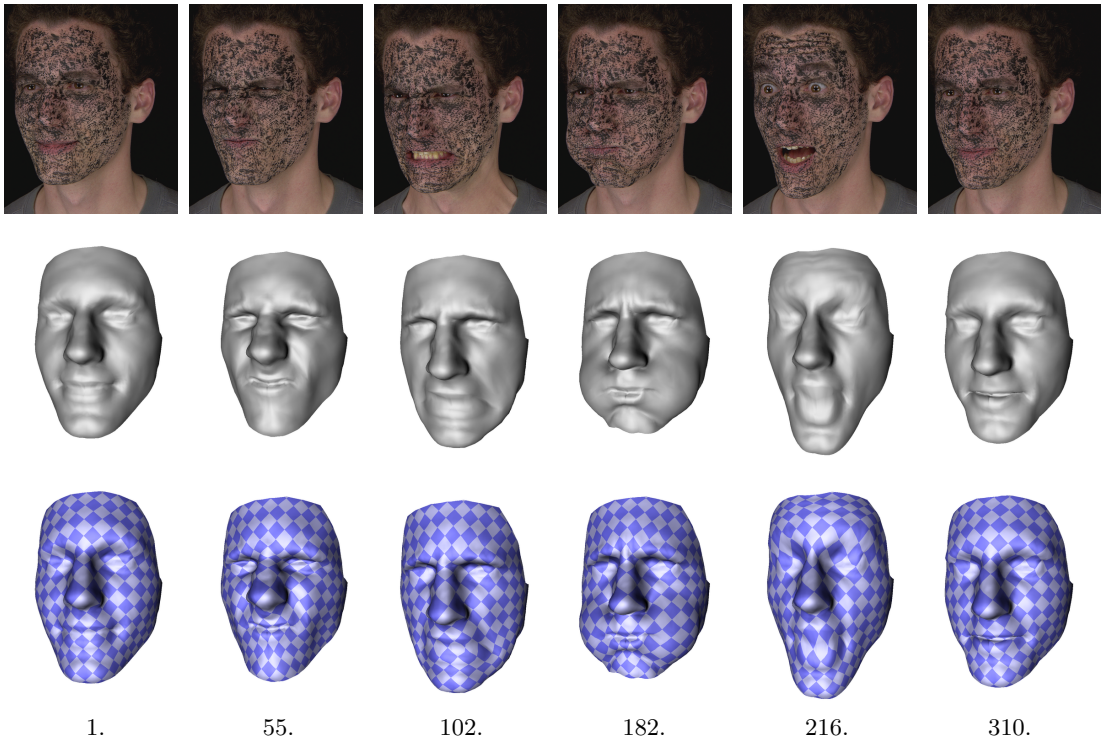


Figure 5.10: Snapshots from the temporally consistent mesh sequence for the dataset Martin-pattern1.

5.5.3 Quantitative evaluation using unwrapped surface textures

The temporal consistency of the aligned mesh sequence $\{M_t\}_{t=1}^{T}$ can be quantitatively evaluated according to stability of facial appearance projected onto M_t at every frame t. The tracked mesh M_r at the initial frame is unwrapped into a 2D planar domain. Given a fixed topology Γ of $\{M_t\}_{t=1}^{T}$ this unwrapping provides a fixed UV domain for time-varying facial textures. At every frame the images $\{I_t^c\}_{c=1}^{C}$ are projected from their views onto the mesh M_t and a single texture for M_t is stored in the UV domain. To avoid introduction of any additional error to the evaluation measure described later, the texture should contain the original image information with minimal alteration and without any time-varying artefacts. Therefore, each half of the face is textured from a single side view and there is no blending between the halves (a seam is visible across the face in the video). The result is a sequence of unwrapped facial textures with the same texture coordinates in 1000×1000-pixel UV domain (note this is a bounding box around the unwrapped mesh and the effective texture area is smaller).

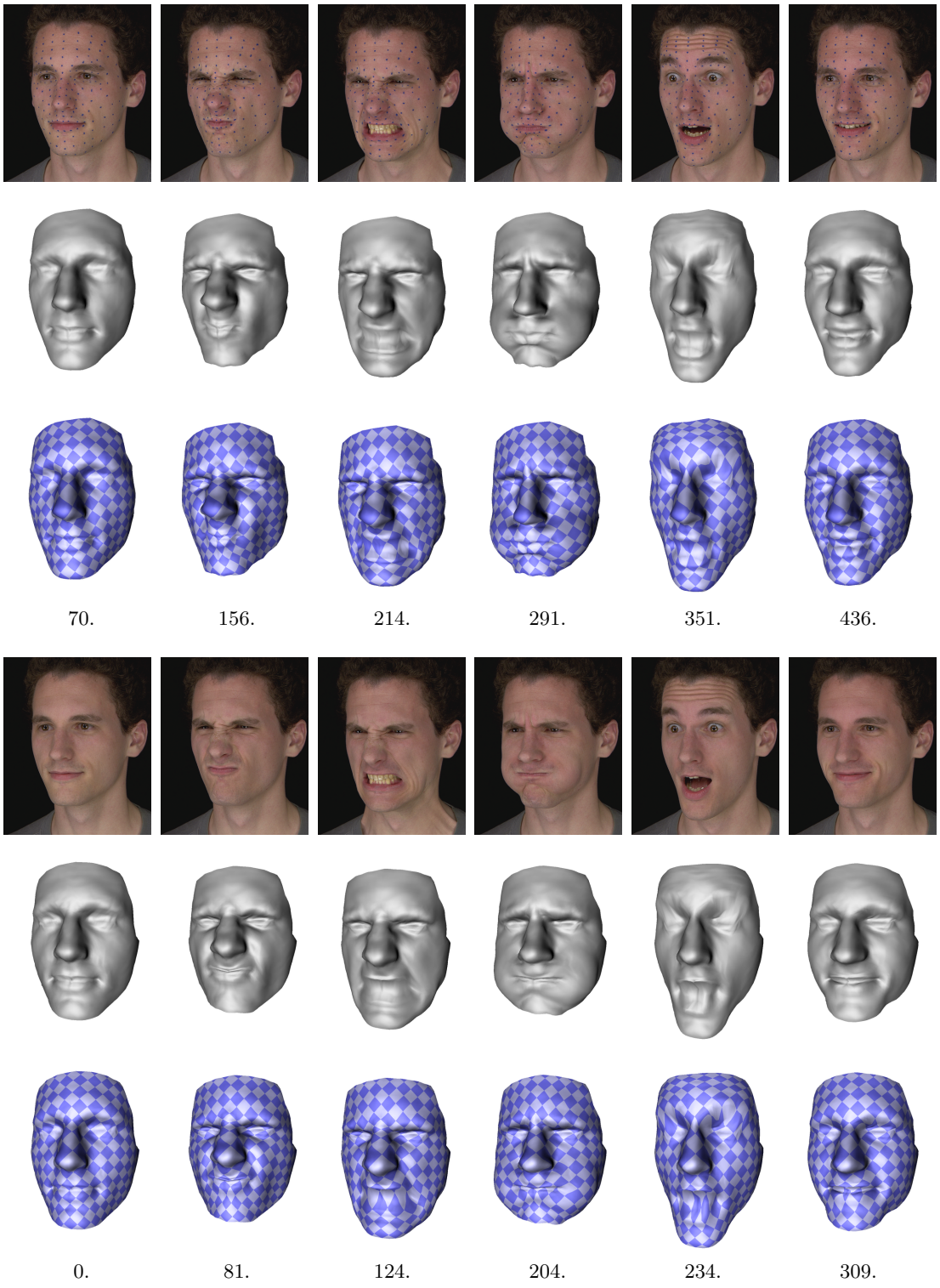


Figure 5.11: Snapshots from the temporally consistent mesh sequences for the datasets Martin-markers1 (top) and Martin-skin1 (bottom).

The facial appearance in the texture sequence should be stable over time if the surface is tracked accurately. Drift in the temporally aligned mesh sequence translates to drift of the texture in the mesh UV domain. Thus, comparison of the texture at the initial frame r with the textures at all other frames should reflect a per-frame quality of temporal alignment. However, this comparison is affected by changes of the facial appearance itself due to skin wrinkling, shading changes, etc. Even with a perfect temporal alignment the texture sequence is not the same throughout a performance. Therefore, any fluctuation of a texture comparison measure around peaks of facial expressions is caused to some extent by the appearance changes. For the comparison between the initial frame and the rest of the frames a sum of absolute differences (SAD) is used. This is computed across all pixels in the effective texture area and differences in RGB channels are added together. Afterwards, the sum is normalised by the number of pixels so that the absolute difference is expressed per pixel. This error measure only expresses a difference between textures and does not directly reflect the true magnitude of drift in the UV domain which is unknown. Optic flow would be a better option from this perspective but issues of the flow algorithms would add a systematic error into the measure.

Figure 5.12(a) shows a profile of per-pixel SAD error for the proposed robust technique and the baseline technique on the dataset Martin-pattern1. The error fluctuates for both techniques with local maxima at extremes of facial expressions which are different from the initial neutral expression. The baseline method recovers after each expression to a similar level of error thanks to the use of reference patch textures from the initial frame. The robust method has generally higher error which increases towards the end of performance. This indicates drift of the mesh which gets worse after every expression. The reason is adaptivity of patch textures which are updated to slightly different points during fast surface deformations. However, a relatively large quantitative difference between the methods maps to qualitatively similar results with a bit larger drift for the robust technique. The nature of the pattern texture where bright and dark areas alternate densely across the face causes that even slight mesh misalignment to have a strong response in SAD. The amount of texture generally influences a magnitude of SAD. Therefore, it is invalid to compare the error values between surfaces with different

textures (the plain skin has always lower values than the pattern because of its uniform texture).

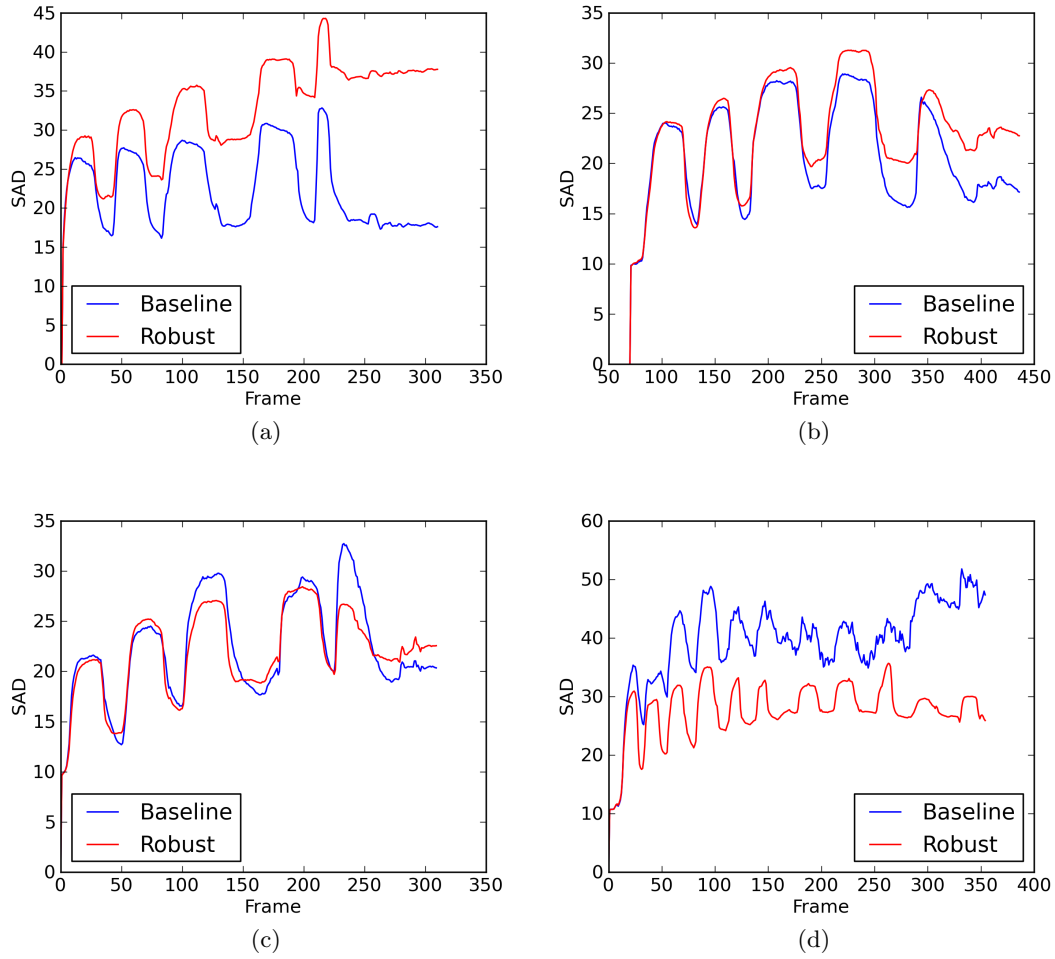


Figure 5.12: Per-pixel SAD error on the sequence of unwrapped facial textures for the datasets Martin-pattern1(a), Martin-markers1(b), Martin-skin1(c) and Martin-skin2(d). The error is summed across RGB channels and the unit is a channel level. Also note that its magnitude cannot be compared across varying surface texture.

The dataset Martin-markers1 exposes drawbacks of the evaluation measure. The robust method produces coherent tracking with a bit of drift across the whole face which gradually builds up towards the end. The baseline method provides accurate alignment of the facial expressions similar to the neutral reference but there are significant wobbles in the unwrapped texture during the other expressions. However, this fact is largely concealed in SAD which is already high at those points due to appearance difference.

This is the reason why the magnitude of error peaks is similar for the both methods in Figure 5.12(b). Moreover, the robust method has a higher error later on because of the small drift in the regions with more stable appearance over time (e.g. cheeks). The comparison to the initial frame has an implicit bias towards it. Even a well-aligned segment of performance which is a bit off from the reference appearance is penalised more than large short-term distortions, although it is perceptually more acceptable.

Despite the error measure favouring the baseline technique because of the kind of errors it introduces, there is a clear quantitative improvement by the robust technique for the plain skin. The baseline technique has significant problems with the less challenging dataset Martin-skin1 (Figure 4.13(bottom)). This is illustrated in Figure 5.12(c) by increasing error and larger fluctuations than the robust method. For the dataset Martin-skin2 the robust technique (Figure 5.13(top))is significantly better in Figure 5.12(d) since the baseline technique fails on fast expression changes. The error profile of the robust approach is higher than in Martin-skin1 due to higher difficulty of the performance.

5.5.4 Variants of frame-to-frame alignment

Several features of the frame-to-frame non-rigid alignment presented in this chapter make surface tracking successful on the face with plain skin. These are cooperative random sampling used for 3D patch matching, adaptive patch textures updated at every frame and shape prior in the form of raw mesh M_t^g constraining the matching. This section compares different variants of the robust method omitting individual features to demonstrate their influence. The omitted features are replaced by their equivalents from the baseline method which shows their importance for accurate surface tracking. Table 5.2 describes configurations of two main methods - Baseline, Robust and additional variants - Robust-IGD, Robust-NoPrior, Robust-FixedTexture.

Evaluation of the variants is performed on the dataset Martin-skin2 which provides more difficult performance than the dataset Martin-skin1. An actor performs at faster pace with a larger variety of facial expressions. The parameters defined in Table 5.1 are the same for all variants of the method. The variants Baseline and Robust-IGD

Variant	optimisation	shape prior	patch texture
Baseline	IGD	no	fixed
Robust	CRS	yes	adaptive
Robust-IGD	IGD	yes	adaptive
Robust-FixedTexture	CRS	yes	fixed
Robust-NoPrior	CRS	no	adaptive

Table 5.2: Variants of frame-to-frame alignment with their configuration of main algorithmic features.

with IGD optimisation do not use the expansion of motion estimates for initialisation (Equation 4.3) because of instability reported in Section 4.5.2. The variant Robust-IGD also does not employ the initialisation by multi-view stereo optimisation (Equation 4.2) because the fitting to M_t^g reconstructed by multi-view stereo has a similar effect. The variant Robust-NoPrior optimises only the image alignment term of Equation 5.1 ($w_g = 0$). The variant Robust-FixedTexture uses the reference patch textures $\{B_i^c\}_{c=1}^C$ in the image alignment term in Equation **??** (same as in Equation 4.1 for the baseline technique).

The best result is achieved by the complete method Robust as can be seen in Figure 5.13(top). However, the difficulty of the dataset Martin-skin2 exposes limits of the method as well. There is noticeable mesh distortion in the most deforming regions (eye sockets, lips) which accumulated over time. Shape details are a bit smoother than for the dataset Martin-skin1 (Figure 5.11(bottom)) because of the stronger regularisation necessary for coping with more challenging motions. In comparison to the dataset Synthetic-skin1 created from the performance Martin-skin2, there is a bit worse temporal alignment in the most problematic regions. But this still proves that the robust method addresses the problem with large skin appearance variation. The method Baseline from the previous chapter performs the worst. Figure 5.14(bottom) depicts major deformations of facial shape and poor temporal alignment.

The variant Robust-IGD improves over the variant Baseline by means of the texture adaptation and the constraint by raw geometry. Figure 5.14(middle) shows more preserved facial shape and less severe distortions. However, IGD still yields significant drift and mesh deformations (e.g. distorted chin at frame 67). CRS proves to be the

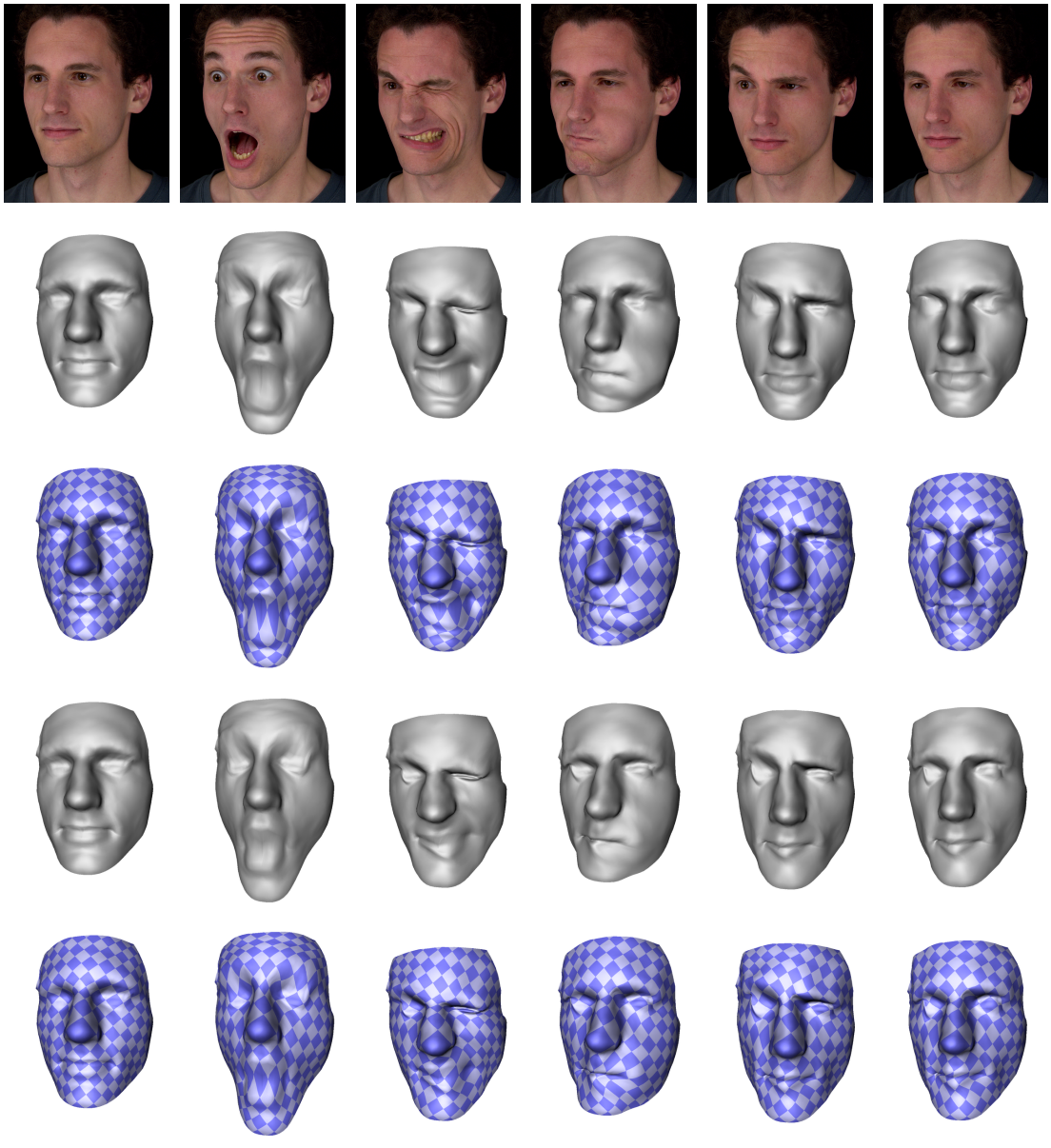


Figure 5.13: Snapshots from the temporally consistent mesh sequences for the dataset Martin-skin2 - the variant Robust (top) and the variant Robust-NoPrior (bottom).

feature with the biggest impact. The variant Robust-FixedTexture achieves even bigger improvement over the variant Baseline (Figure 5.14(top)). However, it still suffers from distortions of the mesh at the peaks of expressions as the baseline method. This is caused by matching the patch textures from the reference frame to very different facial appearance at different expression. In some cases, the tracking is not able to recover and the mesh stays locally distorted (e.g. root of the nose and lips in Figure 5.14(top)). The adaptive textures ensure more coherent motion over time but have less importance than CRS optimisation. The variant Robust-NoPrior provides the results closest to the variant Robust in terms of temporal consistency. But the shape of the face degrades throughout the performance as is visible in Figure 5.13(bottom). This is caused by the patch texture update even if there are alignment errors occurring. A patch can then gradually adapt to a 3D point away from the actual surface. However, this problem is mostly supressed by robust CRS optimisation which finds the global minima even though the search space is not constrained by the unregistered mesh. Thus, the shape prior has the least but still important influence on the quality of resulting mesh sequence.

Figure 5.15 shows a comparison of the algorithmic variants using SAD error on unwrapped facial textures. The variant Baseline is clearly the worst with Robust-IGD also providing low-quality tracking. The variant Robust and Robust-NoPrior have similar profile because the lack of shape prior has the least impact on the robust technique. Also, moderate shape distortions, which are the main difference between their results, do not distort significantly textures projected onto the mesh. According to Figure 5.15 the variant Robust-FixedTexture has the lowest error throughout the performance in contrast to visual assessment of unwrapped textures where significant shakes appear during the peaks of expressions. These shakes are similar to the baseline technique which also uses fixed patch textures. Thus, the SAD measure favours the variant Robust-FixedTexture over Robust for the reasons mentioned before.

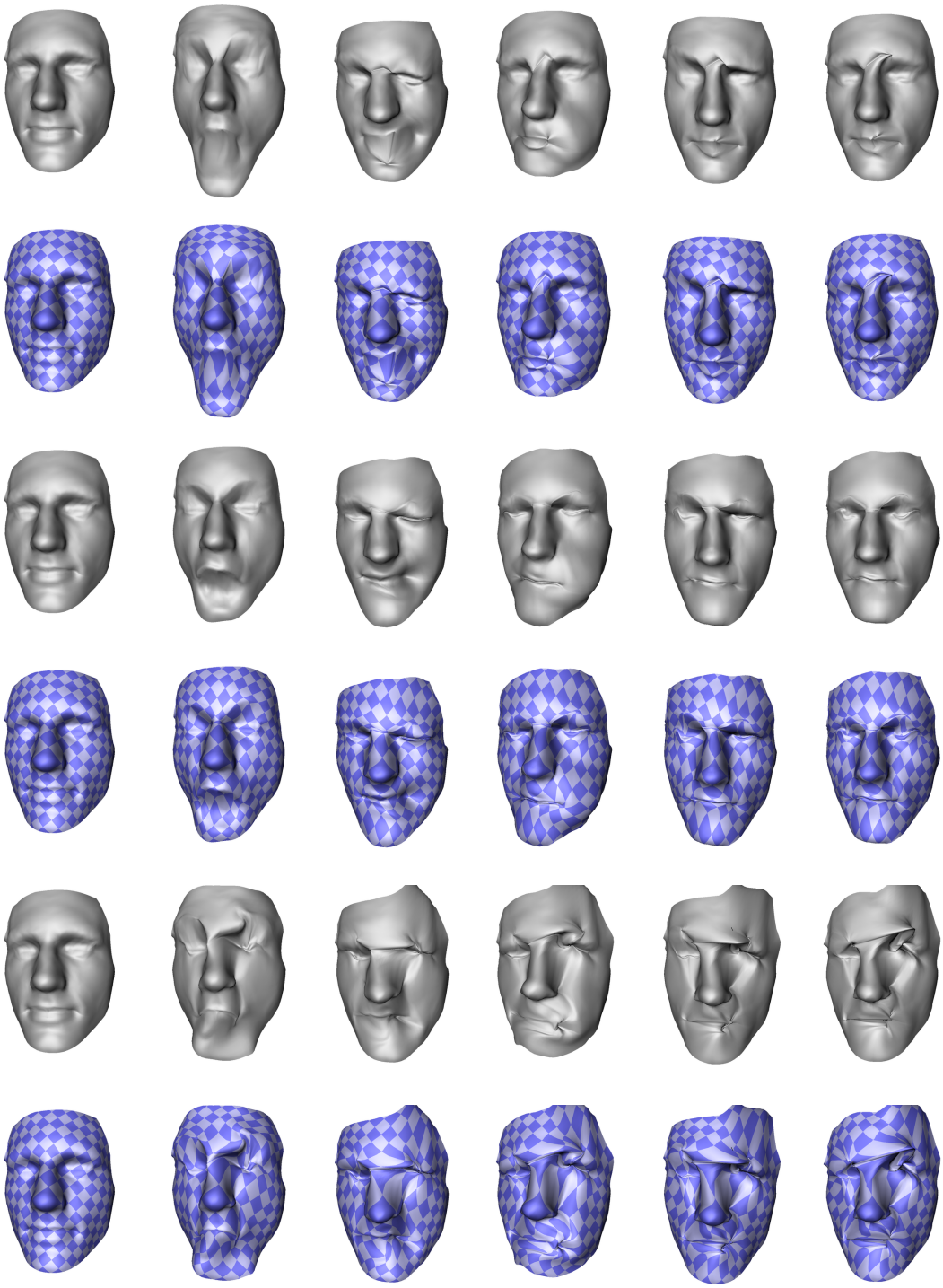


Figure 5.14: Snapshots from the temporally consistent mesh sequences for the dataset Martin-skin2 - the variant Robust-FixedTexture (top), the variant Robust-IGD (middle) and the variant Baseline (bottom).

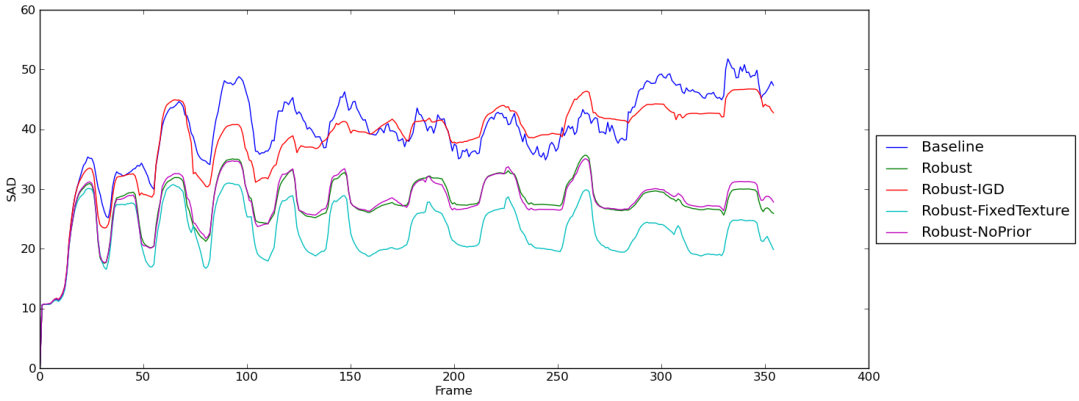


Figure 5.15: Per-pixel SAD error on unwrapped facial textures for the variants of frame-to-frame alignment (the dataset Martin-skin2).

5.5.5 Coarse-to-fine processing

Coarse-to-fine processing using a hierarchical surface model does not improve the final temporally consistent mesh sequence when used with the robust technique (the variant Robust). The aim of coarse-to-fine alignment has been to overcome limitations of the baseline technique (the variant Baseline). However, CRS optimisation combined with other improved features erases benefits of the coarse-to-fine scheme and does not need a more complex hierarchical model. But the coarse-to-fine processing is relevant for the baseline IGD optimisation and an improvement is demonstrated for the method variants Baseline and Robust-IGD on the dataset Martin-skin2.

Both variants work with the same resolution of tracked mesh M_t as their non-hierarchical counterparts. The user-defined mesh M'_r is subdivided the same number of times when the patch surface model is built (yielding 3 LOD). The patch size $N_o = 15$ is decreased by the factor $\psi_o = 0.75$ giving the patches with 15, 11, 8 sample rings across LOD (the largest size on the coarsest LOD). The regularisation is relaxed towards the finest LOD by downscaling the smoothness coefficient s with the factor $\psi_s = 0.5$. The variant Baseline uses $s = 52$ (sequence $52, 36, 13$ across LOD) which higher than $s = 36$ (sequence $36, 18, 9$ across LOD) for the variant Robust-IGD because of lower capabilities.

Figure 5.16(top) illustrates low-quality temporal alignment for the variant Baseline.

However, the mesh sequence is more stable over time and contains less severe shape distortions than the non-hierarchical version (Figure 5.14(bottom)). Clearer improvement can be seen for the variant Robust-IGD in Figure 5.16(bottom). Coarse-to-fine processing helps with fast and large motions such as surprise where IGD struggles on the single mesh resolution (frame 67 in Figure 5.14(middle)). Generally, the result with the coarse-to-fine scheme is more stable and has less distorted mouth and eye regions. Comparing to the variant Robust (Figure 5.13(top)), there is still more local drift created at the peaks of expressions. Therefore, IGD optimisation combined with coarse-to-fine processing performs worse than CRS optimisation.

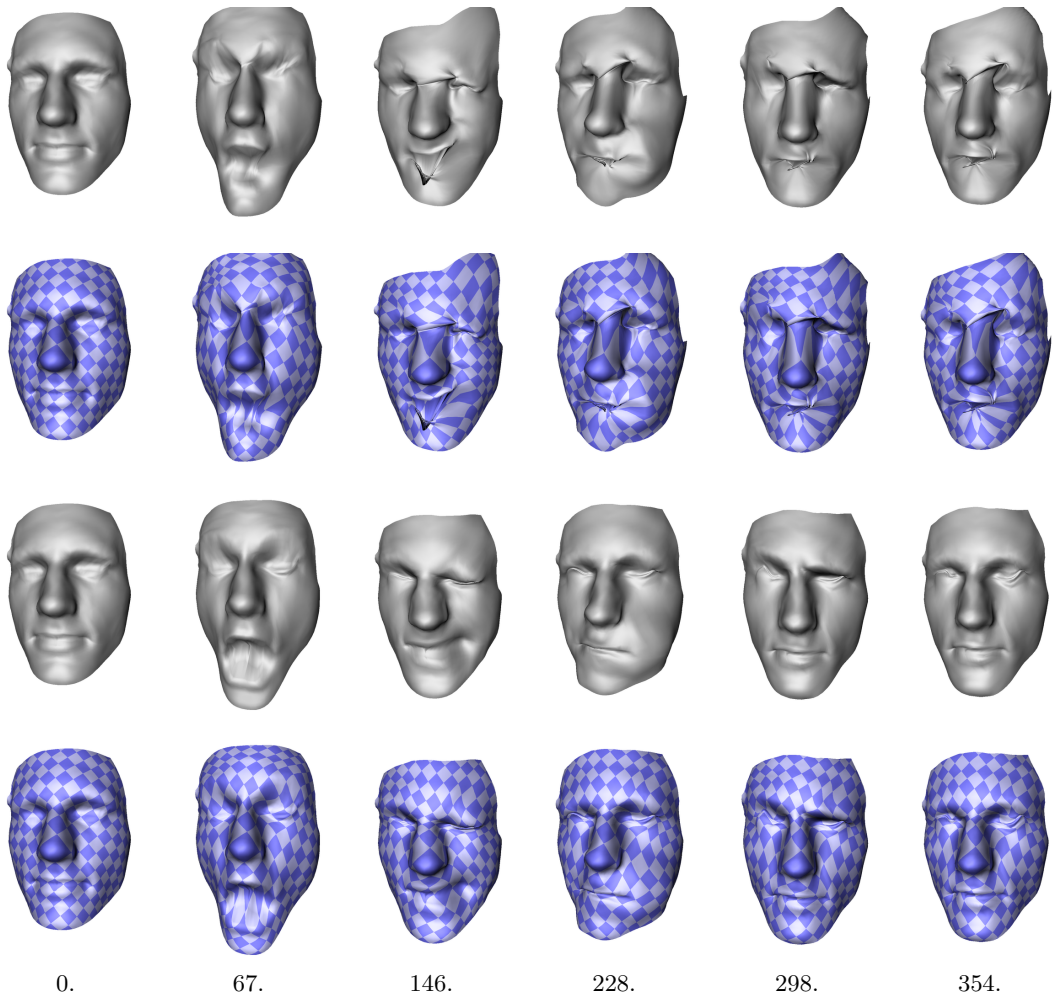


Figure 5.16: Snapshots from the temporally consistent mesh sequences for the dataset Martin-skin2 using coarse-to-fine processing with the variant Baseline (top), the variant Robust-IGD (bottom).

5.6 Conclusion

This chapter has proposed a robust surface tracking method which enables more accurate temporal alignment of facial performances where no markers or pattern are used. This is achieved by introducing several key extensions to the baseline method presented in Chapter 4. Firstly, matching of surface patches is performed by a novel optimisation scheme using cooperative random sampling. Iterative propagation of motion estimates across the surface significantly increases robustness against rapid non-rigid motions and weak surface texture. Also, it is sufficient to optimise only patch position instead of the full six degrees of freedom as for independent gradient descent optimisation. Secondly, the matching objective function is reformulated such that the appearance of the patch from the previous frame is used for alignment with the current frame. Adaptivity of the patch textures helps to overcome instability of the fixed reference textures on expressions with very different appearance from the reference frame. Thirdly, increased risk of drift due to patch texture updating is mitigated by the shape prior in the form of an unregistered mesh sequence reconstructed beforehand. Limiting the patch motion around the unaligned geometry together with adaptive textures makes the objective function less ambiguous in the case of weak surface texture according to the analysis undertaken. This further improves estimation of raw motion vectors by the robust cooperative optimisation, hence accumulation of tracking errors is reduced over longer performances with complex non-rigid deformations. Lastly, a coarse-to-fine approach is proposed for the frame-to-frame alignment of the tracked mesh but this brings marginal benefit combined with the robust cooperative patch optimisation.

The proposed robust approach has been evaluated on a synthetic facial performance and several real performances with varying amount of make-up on an actor's face. Comparison to the baseline technique shows slightly more drift but still accurate tracking for a well-textured surface (a face painted with a random pattern) or a surface without time-varying appearance (a synthetic face with a fixed texture). There is a significant improvement over the baseline for a weakly-textured surface (the face with plain skin), especially when undergoing fast, complex deformations. This demonstrates a clear advance over previous sequential motion capture methods such as [23, 79, 36, 37]

on surfaces with a weak texture. Several variants of the frame-to-frame alignment have been compared to identify the importance of individual extensions. The cooperative patch optimisation provides the largest improvement followed by adaptive patch textures and shape prior with the least but still important influence. The coarse-to-fine processing proves to be redundant combined with the cooperative random sampling, but it is beneficial for the baseline independent gradient descent optimisation. Apart from visual assessment of the results quantitative evaluation has been conducted in spite of the lack of standard evaluation measures for surface tracking. Two error measures are used: correlation of the projected textures in the UV space of the tracked mesh over time; and mesh distance between corresponding frames of mirrored input sequence (Appendix B). Both error measures have their limitations, but can be used for approximate quantitative evaluation of tracking results.

The main drawback of the proposed method remains the accumulation of alignment errors while tracking sequentially through an input sequence. Smoothing of shape details and local drift in the most deforming regions such as the eye sockets and the lips gradually appear over time. These artefacts become significant after long chains of frame-to-frame alignments due to robustness of the proposed method but they are inevitable because of sequential traversal. The next chapter addresses this problem by introducing non-sequential tracking which aligns the input sequence along multiple paths of reduced length.

Chapter 6

Non-sequential surface tracking

The previous chapter presented a new robust technique for surface tracking which handles fast non-rigid deformations of plain skin. Despite the increased robustness of frame-to-frame alignment, there is still drift over longer periods of time in areas with the most deformation. This is due to the nature of sequential tracking which chains frame-to-frame alignments throughout an input sequence. Sequential traversal is also prone to failure of the alignment method which prevents tracking of subsequent frames.

Non-sequential methods for surface tracking have been proposed which reorder the input sequence to alleviate the problems of drift and failure. Beeler et al. [13] identify anchor frames similar to the neutral expression in a reference frame across a facial performance using direct image correlation. Initial alignments are made from the reference frame to the anchor frames and the segments between anchors are tracked sequentially in both directions. The result at each frame is selected from two hypotheses according to an estimate of tracking error. Tracking of the facial mesh is based on per-frame 3D reconstructions and multi-view optic flow. Global alignment of multiple unregistered mesh sequences of whole-body performance is tackled by Huang et al. [52]. The frames across all sequences of the same actor are compared using a dissimilarity measure based on the shape of unregistered meshes. Their comparison is performed through difference of shape histograms as recommended in [54]. Sequences are linked through a few pairs of the most similar frames. Each sequence and the links between them are sequentially tracked by a geometry-based alignment method. The shortest path tree is

calculated among frames using the dissimilarity to establish the optimal way of combining the tracking results. A mesh with fixed topology is propagated according to the tree through the sequences using the computed temporal correspondence. This yields a temporally consistent representation across the multiple input mesh sequences. In contrast, Budd et al. [22] optimise the traversal among all frames of several whole-body performances based on the minimum spanning tree. This is calculated on a fully connected graph among all frames using the shape-histogram dissimilarity. Therefore, the minimum spanning tree minimises the total path through the whole shape dissimilarity space rather than the sub-space limited to a few non-sequential links as in [52]. The traversal of the input data is then less sequential and also directly guides the actual surface tracking in contrast to [52].

The approach using the minimum spanning tree [22] has been chosen as a basis for the non-sequential tracking framework presented in this chapter. The concept is for the first time used in the scenario of facial performance capture. This requires a different dissimilarity measure which is more suitable for relatively subtle motion of the face than the shape histogram. Thus, calculation of the minimum spanning tree is based on the dissimilarity derived from the motion of a sparse set of strong facial features. The tree then guides the robust frame-to-frame alignment method from the previous chapter throughout the input sequence. Although, the tree traversal provides shorter alignment paths and thus less drift, it can suffer from alignment inconsistencies where different paths meet. The independent accumulation of errors along different paths can lead to jumps in the final mesh sequence.

It has been observed that there is a trade-off between drift and jump errors. Therefore, a novel cluster tree is proposed to balance these two types of errors. The cluster tree enforce sequential tracking inside clusters of similar successive frames, but still use non-sequential transitions between them based on the minimum spanning tree. Granularity of the frame clustering influences the shape of the tree which allows selection of the optimal tree for a given input sequence. Although, the cluster tree reduces the number of places where different alignment paths meet, the jumps may still occur. Fusion of tracking results across tree branches is proposed to eliminate the jumps in the temporally consistent mesh sequence.

This chapter presents a formulation of the non-sequential surface tracking and calculation of several types of traversal trees (minimum spanning tree, shortest path tree, cluster tree). Characteristics of optimal traversal tree are discussed from a theoretical perspective. The proposed surface tracking framework is generalised to any frame-to-frame alignment method with an associated dissimilarity measure. It is not limited to the face application and comprehensive evaluation shows the suitability for other non-rigid surfaces such as cloth and whole body. Two different frame-to-frame alignment methods with respective dissimilarity measures are investigated to demonstrate wider applicability of the framework. The image-oriented method proposed in Chapter 5 is combined with the dissimilarity from sparse features for facial performances and cloth deformation. The geometry-oriented method from [22] is combined with the dissimilarity from the shape histogram for whole-body performances. The validity of the dissimilarity measures used is empirically assessed for their respective alignment methods.

6.1 Problem formulation

The input is a sequence of observations $\{O_t\}_{t=1}^{T}$ of a deforming surface for frames $\{1, ..., T\}$ where O_t is defined in Section 5.1. The sequence can consist of multiple segments from independent motions of the surface and T is then the total number of frames in all segments. Conventionally, a temporally consistent mesh sequence $\{M_t\}_{t=1}^{T}$ is obtained by *sequential tracking* which concatenates *frame-to-frame non-rigid alignment* between successive frames $t-1$ and t. The frame-to-frame alignment estimates the correspondence between observations O_{t-1} and O_t. Any errors in the correspondence influence subsequent alignments which leads to drift in $\{M_t\}_{t=1}^{T}$.

Non-sequential tracking processes the input sequence $\{O_t\}_{t=1}^{T}$ in an order different from the temporal order. A traversal of $\{O_t\}_{t=1}^{T}$ is guided by a measure which estimates the difficulty of non-rigid alignment of observations O_t between any pair of frames. Intuitively, the difficulty of transition between frame i and j is represented by a *dissimilarity* d between respective observations O_i and O_j (example measures are described in Section 6.2). Given a symmetric measure d between all pairs of frames, alignment paths

to every frame are jointly optimised to have minimal length according to the dissimilarity. The tracking is then performed along multiple paths of shorter length than fully sequential traversal which reduces the accumulation of alignment errors.

A traversal of the input sequence is represented by a *traversal tree* $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ which is a spanning tree with the nodes $\mathcal{N} = \{n_i | \forall i \in [1..T]\}$ corresponding to all frames (Figure 6.1). The directed edge set $\mathcal{E} \subseteq \{(n_i, n_j) | \forall i, j \in [1..T]; i \neq j\}$ has the size $|\mathcal{N}| - 1$ and connects all nodes. The edges are weighted by d between individual observations O_i and O_j. The tree $\mathcal{T}$ has a defined root node n_r which sets directionality of the edges towards the leaf nodes. An alignment path $n_r \rightarrow n_i$ for the frame i starts at n_r and follows the tree structure towards n_i. Non-sequential tracking of the whole sequence chains frame-to-frame alignments from n_r along tree branches towards all leaf nodes. Tracking using a tree leads to the presence of *cuts* in the sequence at places where two different alignment paths meet (marked red in Figure 6.1). Independent accumulation of alignment errors along these paths can potentially manifest as jumps or glitches in the resulting sequence $\{M_t\}_{t=1}^{T}$. Consequently, there is a trade-off between the minimisation of alignment path length and a number of cuts. Longer paths lead to larger gradual drift but a large amount of cuts results in many jumps and jitter. The proposed method reflects this trade-off and allows calculation of the traversal tree which balances between these two kinds of artefacts.

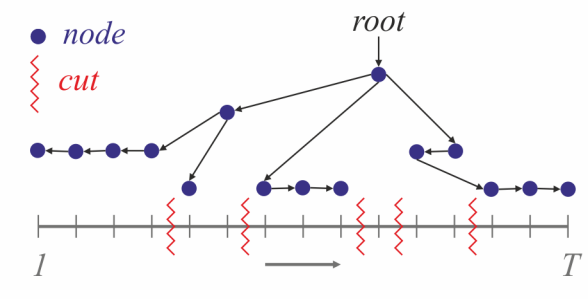


Figure 6.1: A traversal tree $\mathcal{T}$ on the input frame sequence $\{1, ..., T\}$. The cuts separate adjacent frames which have different alignment paths along tree branches.

The non-sequential traversal of the input sequence using $\mathcal{T}$ can be combined with any frame-to-frame surface alignment technique working with $\{O_t\}_{t=1}^{T}$. For calculation of $\mathcal{T}$ it is assumed that the dissimilarity measure d is proportional to the alignment error

of the technique used. However, d is designed as an approximate measure which is significantly easier to compute than actual alignment of the mesh M. The relationship between d and the alignment error is usually non-linear to some extent (Section 6.6.6). Low dissimilarity values map to a similar level of alignment quality. The mid-range is proportional to the alignment error. Above a certain dissimilarity value the alignment algorithm starts to fail and the errors become disproportionately large. This non-linearity partially biases the calculation away from the optimal tree $\mathcal{T}$ for the frame-to-frame alignment method.

6.2 Dissimilarity measure

Image-oriented frame-to-frame alignment proposed in Chapter 5 is coupled with the dissimilarity measure d_I. The dissimilarity is based on a sparse set of surface points which approximates the overall motion of a surface. The motion estimation for the small amount of points is significantly faster than the full frame-to-frame alignment of the mesh. The points are manually chosen by a user and they are tracked in the 3D space throughout the sequence. At first, each surface point is sequentially tracked in one of the image sequences $\{\{I_t^c\}_{c=1}^C\}_{t=1}^T$ where it is the best visible. Afterwards, 2D trajectories of the points are back-projected from the respective views onto unregistered geometries $\{M_t^g\}_{t=1}^T$. This yields their 3D trajectories over time which represent a sparse motion field of the surface.

Although, the sparse surface motion is derived by a sequential tracking process, this does not introduce a bias into subsequent computation of a non-sequential traversal. The reason is Linear predictor tracker [81] used for the 2D tracking which has the ability to recover from errors, hence suffers from a limited drift. The method is also robust against a weak texture and complex non-rigid deformations which allows accurate motion estimation particularly in the face application. These properties are achieved by learning appearance variance of the point and creating a specific tracker. This requires manual landmarking of the point at distinct surface poses during its motion (in the case of faces these are extremes of expressions and midpoints between them). The number of landmarked frames depends on the complexity of the observed motion but it is a

small fraction out of the whole sequence. The tracker is trained on examples from different time instances which provides the important additional capability in comparison to standard sequential trackers without any prior learning. After training phase, the actual tracking of the point is fully automatic with the ability to recover from moderate failures. Additionally, the learned tracker can be reused for different sequences of the same surface as long as the variety of motion is similar to the original sequence.

The dissimilarity d_I between two frames is defined as an average Euclidean distance between 3D positions of the tracked points at these frames. This assumes that the spatial difference between the point configurations indicates the difficulty of full mesh alignment. To encourage transitions between similar instances of the surface with just a different global pose, the sets of 3D positions are rigidly aligned before the comparison. This is performed by Ordinary Procrustes analysis [34], which minimises the Euclidean distance between the corresponding points using least squares. The resulting rigid transformation initialises the frame-to-frame non-rigid alignment algorithm during the actual tracking. In the case of facial performance, this approach effectively discards the influence of head pose and the traversal is based on facial expressions only.

The dissimilarity d_I is an approximate measure so it does not have to accurately represent the surface motion. The scores typically contain imprecisions due to errors in 2D tracking or artefacts in the unregistered meshes. Also, it is influenced by the number of surface points used and their individual motion variations. A large number of points with limited motion can dilute the influence of distinct ones and flatten the resulting dissimilarity matrix among frames. On the other hand, a few points with a distinct motion can encompass smaller amount of motion nuances which can lead to worse discrimination between relatively similar frames. Furthermore, the prior rigid alignment can be inaccurate if the number of points used is too small. Typically, a small set of points is sufficient (for example $15-20$ for the face). Despite more complex calculation, the measure d_I is more valid than direct image correlation among frames used by Beeler et al. [13]. The correlation does not directly reflect the amount of surface motion in the 3D space.

Another example of a dissimilarity measure is proposed by Budd et al. [22] for whole-body performance tracking. An unregistered sequence of body meshes $\{M_t^g\}_{t=1}^T$ is non-sequentially aligned using a geometry-oriented frame-to-frame method. A traversal tree is determined by the dissimilarity d_G which makes use of a shape histogram [54]. This is a volumetric histogram using a spherical partitioning of the 3D space which encodes the surface shape. The dissimilarity d_G between frames i and j is SSD between the histograms for meshes M_i^g and M_j^g. The position and orientation of the spherical grids for both histograms are optimised to minimise SSD. A side product of the optimisation is a rigid alignment between the meshes which discards different overall pose. This transformation is used to initialise the frame-to-frame non-rigid alignment. The measure d_G does not directly describe the surface motion between frames, but a change in the surface shape. There is an implicit assumption that motion is generally associated with shape changes. This is not correct for motions such as shrinking or stretching but these are not common for the whole body movement.

6.3 Traversal tree

A traversal tree $\mathcal{T}$ for non-sequential tracking can have different forms. Fully sequential traversal is the special case of tree with a single branch starting at the initial frame. The following sections present several types of the traversal tree - minimum spanning tree, shortest path tree and cluster tree which are calculated according to different objective functions.

6.3.1 Minimum spanning tree

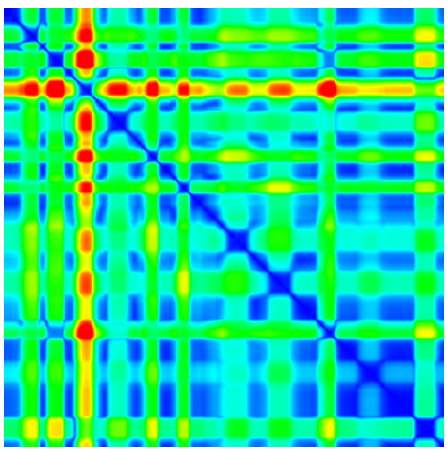
A non-sequential traversal of an input sequence based on the *minimum spanning tree* (MST) has been introduced by Budd et al. [22]. It is computed in a dissimilarity space based on shape histogram comparison and used for geometric alignment of unregistered mesh sequence. This concept is generalised here for an arbitrary dissimilarity d between multi-modal observations O_t at every frame. The space of all possible pair-wise transitions between frames of the sequence is represented by a dissimilarity matrix $\mathbf{D}$ of size

$T \times T$ where both rows and columns correspond to individual frames (Figure 6.2(a)). An element $\mathbf{D}(i,j)$ is a dissimilarity d between O_i and O_j which defines a cost of alignment between frames i and j. The matrix $\mathbf{D}$ is symmetric and has zero diagonal. The optimal traversal in this space can be found through graph formulation of the problem as suggested in [22].

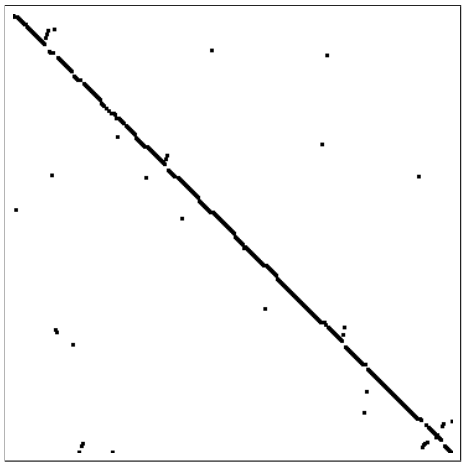
A fully-connected undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{D})$ is built from the matrix $\mathbf{D}$. The nodes $\mathcal{N}$ are associated with all frames and the edges $\mathcal{D} = \{(n_i, n_j) | \forall i, j \in [1..T]; i \neq j\}$ have the weight $\mathbf{D}(i,j)$. A traversal visiting all frames is described by an undirected spanning tree $\mathcal{T}' = (\mathcal{N}, \mathcal{E}')$ where $\mathcal{E}' \subset \mathcal{D}$. The optimal tree $\mathcal{T}'_{MST}$ is defined as the *minimum spanning tree* (MST) which minimises the total cost of pair-wise alignment given by d as outlined in Equation 6.1. This objective approximates the total non-rigid deformation of the surface which has to be overcome following the traversal tree.

$$\mathcal{T}'_{MST} = \underset{\mathcal{T}' \subset \mathcal{G}}{\operatorname{argmin}} \left(\sum_{(n_i, n_j) \in \mathcal{T}'} \mathbf{D}(i,j) \right) \tag{6.1}$$

Equation 6.1 is optimised using Prim's algorithm [88]. Note that MST does not define a root node, thus it has to be selected to set directions of the traversal. The tree $\mathcal{T}_{MST}$ is the directed version of $\mathcal{T}'_{MST}$ with the optimal root node selected by Equation 6.8.



(a)



(b)

Figure 6.2: The dissimilarity matrix $\mathbf{D}$ for the dataset Synthetic-skin1 (blue - low values, red - high values) (a). The traversal tree $\mathcal{T}_{MST}$ depicted in $\mathbf{D}$ (each directed edge (n_i, n_j) is marked black at respective location $\mathbf{D}(i,j)$) (b).

The benefit of MST is that low-cost transitions are at the centre of the tree because of their priority during the tree construction. The edges with larger d are therefore pushed towards the leaf nodes. This reduces the accumulation of errors along the branches and also limits the extent of failure due to large inter-frame dissimilarity to the ends of branches. The drawback of MST is that it does not take into account the introduction of cuts and tends to temporally over-fragment the sequence. This is depicted in Figure 6.1 of a generic traversal tree $\mathcal{T}$. Notice temporal reshuffling of frames on the single branch on the right and the short offshoot on the leftmost branch. This happens mostly in slow-motion segments where $\mathcal{T}_{MST}$ over-fits on small changes of the surface (notice in the lower right corner of Figure 6.2(b)). The fluctuation of d values in these segments can also be a consequence of inaccuracies in the dissimilarity measure which is approximate by design. The sensitivity of MST to noise in d is not desirable. Tracking results obtained using MST can contain many noticeable jumps throughout the sequence and jitter of the mesh during static poses or slow motions.

6.3.2 Shortest path tree

The minimum spanning tree minimises the total dissimilarity between frames which has to be overcome during the non-sequential tracking. However, this objective does not explicitly optimise path dissimilarity lengths to individual frames which indicate potential error accumulation from frame-to-frame alignments. The *shortest path tree* (SPT) used in [52] minimises directly the path length to all nodes from the selected root which reflects total amount of potential drift at each frame. $\mathcal{T}_{SPT}$ is a directed tree, in contrast to $\mathcal{T}_{MST}$, calculated from the same graph $\mathcal{G}$. Equation 6.2 defines the optimisation of a directed spanning tree $\mathcal{T}$ with a given root node n_r according to SPT criterion,

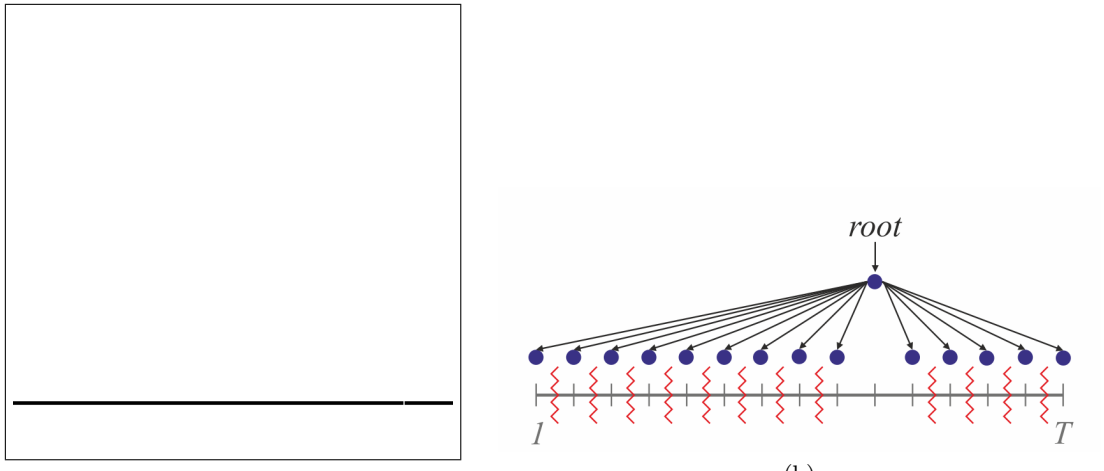
$$\mathcal{T}_{SPT} = \underset{\mathcal{T} \subset \mathcal{G}}{\operatorname{argmin}} \left(\sum_{n_t \in \mathcal{T}} \sum_{(n_i, n_j) \in n_r \to n_t} \mathbf{D}(i, j) \right) \tag{6.2}$$

where $n_r \to n_t$ is a path between the root n_r and an arbitrary node n_t. This is computed by Dijkstra's algorithm [30]. The tree $\mathcal{T}_{SPT}$ is optimal with respect to the given root n_r, thus the best n_r has to be selected to obtain the final tree for the whole sequence. Equation 6.2 is evaluated for all nodes taken as the root and the tree with the lowest

total path length is selected as the optimal $\mathcal{T}_{SPT}$.

SPT favours a large number of short branches with relatively high dissimilarities in comparison to the MST. Thus, it introduces many cuts in the sequence which increases the risk of alignment inconsistencies. Edges with relatively high dissimilarities often lead to gross tracking errors because the dissimilarity measure does not reflect well that the alignment algorithm fails above a certain level of dissimilarity between frames. Therefore, MST provides better traversal by prioritising low-dissimilarity transitions and creating less cuts.

In practice, SPT produces the extreme case where there are direct edges from the root node to all other nodes as illustrated in Figure 6.3(b). This is because the evaluation datasets do not contain very high dissimilarities in their matrices which would make branches with multiple edge more optimal in some cases. Figure 6.3(a) demonstrates the single-step SPT for the dataset Synthetic-skin1 by a single line at the row of the root node in the dissimilarity matrix. The aligned mesh sequence using this tree suffers from a lot of jitter because there are cuts between all adjacent frames. Also many gross errors are present because of edges with high dissimilarity which lead to alignment failures.



(a) (b)

Figure 6.3: The traversal tree $\mathcal{T}_{SPT}$ depicted in $\mathbf{D}$ for the dataset Synthetic-skin1 (a). An example illustration of $\mathcal{T}_{SPT}$ on the input sequence (b).

6.3.3 Cluster tree

Both MST and SPT introduce many cuts into the input sequence reordering which results in jumps and jitter in the temporally consistent meshes $\{M_t\}_{t=1}^{T}$. This is caused by the fact that these tree algorithms do not consider the temporal order of frames, thus they are not aware of introducing the cuts. To illustrate, if the input sequence $\{O_t\}_{t=1}^{T}$ is randomly shuffled, the weights of edges in the fully-connected graph $\mathcal{G}$ do not change, and therefore the trees $\mathcal{T}_{MST}, \mathcal{T}_{SPT}$ are calculated the same way.

To address shortcomings of MST and SPT the notion of temporal order of frames needs to be incorporated into the algorithm generating the traversal tree. A novel *cluster tree* is proposed which enforces sequential tracking locally to reduce fragmentation of the sequence. Sequential tracking is favoured in slow-motion segments where the non-sequential traversal does not bring any benefit due to low difficulty of frame-to-frame alignments. This addresses over-fragmentation in these segments by both MST and SPT. The MST approach is still used to link the sequential segments together to obtain global non-sequential traversal of the sequence. Therefore, high-dissimilarity transitions are not likely to be included as in the case of SPT. The cluster tree shape is simpler with a smaller number of cuts which reduces the jumps and jitter in favour of relatively smooth sequential drift which is perceptually more acceptable.

Increased sequential traversal can also alleviate issues with possible inaccuracy of the approximate dissimilarity measure. Noise observed in nearly-static segments of the surface motion does not influence the tree shape because sequential tracking is enforced. General bias in tree calculation due to the non-linear relationship between the dissimilarity and capabilities of the alignment method is also mitigated. Sequential tracking over low dissimilarity values effectively assumes their mapping to the same level of alignment difficulty which often the case in practice. On the other hand, high dissimilarity transitions are generally avoided by MST between sequential sub-sequences which reduces risk of complete alignment failure often associated with high dissimilarities.

Frame clustering

Intuitively, segments traversed sequentially should contain little or no deformation of the surface, so there is a minimal accumulation of errors. Clusters of similar successive frames form blocks with low d around the diagonal in the matrix $\mathbf{D}$ (Figure 6.2(a)). Ideally, large clusters should be generated in slow-motion segments and small clusters (even down to individual frames) in the segments with significant surface motion. The summarisation method by Huang et al. [53] is modified for the purpose of frame clustering. The clusters do not have any representative key-frames but all frames are compared to each other to measure overall intra-cluster consistency. This provides a more general clustering approach which suits our purpose better than grouping frames around a few distinct exemplars.

A sequence of frames $\{1, ..., T\}$ can be represented by a clustering $U = \{F_k | \forall k \in [1..K]\}$ where K is a number of non-overlapping clusters. A frame cluster F_k is a sequence of frames $\{t_k - \Delta t_k, ..., t_k + \Delta t_k\} \subset \{1, ..., T\}$ where t_k is a central frame and Δt_k is a half-size of the cluster. The clustering U has to fulfil following conditions: $\bigcup_{k=1}^{K} F_k = \{1, ..., T\} \quad \wedge \quad F_k \cap F_l = \emptyset; k \neq l$. The inconsistency of frames within cluster $A(F_k)$ is defined in Equation 6.3 as the sum of pair-wise dissimilarities which is the main difference to [53].

$$A(F_k) = \frac{1}{2} \sum_{i=t_k-\Delta t_k}^{t_k+\Delta t_k} \left(\sum_{j=t_k-\Delta t_k}^{t_k+\Delta t_k} \mathbf{D}(i,j) \right) \tag{6.3}$$

The clustering U is described by two costs: total intra-cluster inconsistency for all clusters and the number of clusters K. The costs are weighted against each other by the parameter $\beta \in \langle 0, 1 \rangle$ to provide a combined cost which is minimised as:

$$U_\beta = \underset{U}{\operatorname{argmin}} \left(\beta K + (1 - \beta) \sum_{F_k \in U} A(F_k) \right) \tag{6.4}$$

The optimal set of clusters U_β for the matrix $\mathbf{D}$ depends on β which influences granularity of the clustering (Figure 6.4(a)). Values closer to 1 return a smaller number of large clusters and values closer to 0 return a larger number of small clusters.

For a given β Equation 6.4 is solved through a graph-based formulation. A directed graph is built in the space of all possible frame clusters where each node represents a cluster of consecutive frames. Edges link only the nodes for which clusters are temporally adjacent to each other. The clusters containing the first frame are connected to the source node and the clusters containing the last frame to the sink node. The direction of all edges is forward in time. The weight of an edge is determined according to the target cluster F_k: $\beta + (1 - \beta)A(F_k)$. The shortest path from the source to the sink found by the Dijkstra's algorithm [30] minimises Equation 6.4. The resulting chain of nodes represents the optimal clustering U_β for the matrix $\mathbf{D}$ which is given by the parameter β.

Tree calculation

A non-sequential traversal can be computed on the sequence of clusters instead of the original frame sequence using MST as described in Section 6.3.1. The dissimilarity matrix $\mathbf{D}$ is collapsed to a cluster dissimilarity matrix $\mathbf{D}_F$ of size $K \times K$ where rows and columns correspond to the individual clusters from U_β (Figure 6.4(b)). Equation 6.5 defines the dissimilarity $\mathbf{D}_F(k, l)$ between the clusters F_k and F_l as the minimal cost of transition between the respective clusters in the full matrix $\mathbf{D}$.

$$\mathbf{D}_F(k, l) = min(\mathbf{D}(i, j)) \qquad \forall i \in F_k, \forall j \in F_l \tag{6.5}$$

A cluster pair (F_k, F_l) is then linked by the pair of frames (i, j) with minimal dissimilarity $((F_k, F_l) \sim (i, j))$. The matrix $\mathbf{D}_F$ is symmetric with zero diagonal elements as for $\mathbf{D}$ (Figure 6.3(b)). A fully-connected graph $\mathcal{G}_F = (\mathcal{N}_F, \mathcal{D}_F)$ with nodes corresponding to the clusters F_k is built from $\mathbf{D}_F$. An undirected spanning tree which minimises the total cost of transitions among the clusters is calculated as in Equation 6.1. The resulting minimum spanning tree $\mathcal{T}'_F = (\mathcal{N}_F, \mathcal{E}'_F)$ has the same properties as $\mathcal{T}'_{MST}$. The tree with edges $\mathcal{E}'_F \subset \mathcal{D}_F$ is illustrated in Figure 6.4(c).

Afterwards, the tree among clusters $\mathcal{T}'_F$ needs to be transformed to a full spanning tree $\mathcal{T}'_\beta = (\mathcal{N}, \mathcal{E}')$ interconnecting all frames $\{1, ..., T\}$. The set of edges $\mathcal{E}'$ consists of two edge groups $\mathcal{E}'_1$ and $\mathcal{E}'_2$. Firstly, $\mathcal{E}'_1$ is constructed from sparse links $(F_k, F_l) \sim (i, j)$

interconnecting the original clusters at frame level. Equation 6.6 transforms the edges $\mathcal{E}'_F$ between clusters into their respective frame-to-frame edges $\mathcal{E}'_1$ where nodes $n_k, n_l \in \mathcal{N}_F$ and $n_i, n_j \in \mathcal{N}$.

$$\mathcal{E}'_1 = \{(n_i, n_j) | \forall (n_k, n_l) \in \mathcal{E}'_F; (F_k, F_l) \sim (i, j)\} \tag{6.6}$$

Secondly, $\mathcal{E}'_2$ links the rest of the frames within the clusters to $\mathcal{T}'_\beta$. Because of low intra-cluster dissimilarity of frames sequential traversal is enforced among them. Thus, Equation 6.7 defines chains of frames in temporal order for all clusters.

$$\mathcal{E}'_2 = \bigcup_{F_k \in U_\beta} \{(n_i, n_j) | i, j \in F_k; |i - j| = 1\} \tag{6.7}$$

The construction of $\mathcal{T}'_\beta$ does not strictly create cuts at all boundaries between the clusters. Often, the minimal transition between temporally adjacent clusters is the one linking the last frame of the first cluster to the first frame of the second cluster. Therefore, the algorithm has an option to chain together several neighbouring clusters into a single sequential segment if it is deemed optimal.

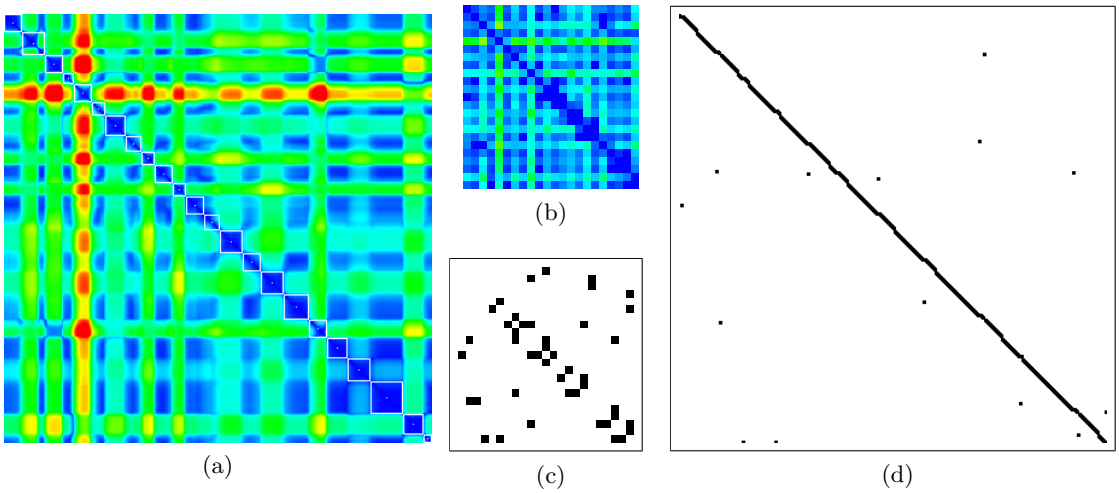


Figure 6.4: A clustering U_β illustrated in the matrix $\mathbf{D}$ for the dataset Synthetic-skin1 (white squares mark individual clusters) (a). A cluster dissimilarity matrix $\mathbf{D}_F$ created by collapsing $\mathbf{D}$ according to U_β (b). A tree $\mathcal{T}'_F$ among clusters is calculated from $\mathbf{D}_F$ (c) and then is expanded into full traversal tree $\mathcal{T}_\beta$ (d). Notice less fragmentation and longer sequential segments than for $\mathcal{T}_{MST}$ (Figure 6.2(b)) or $\mathcal{T}_{SPT}$ (Figure 6.3(a)).

The tree $\mathcal{T}'_\beta$ does not exactly define a traversal of the input sequence because it is undirected and has no root node (similar to $\mathcal{T}'_{MST}$). The root node n_r has to be selected to set directions along the paths in $\mathcal{T}'_\beta$. The selection is made by minimisation of Equation 6.8 which is derived from the criterion for SPT (Equation 6.2). The length of weighted paths $n_u \to n_t$ from a candidate root node n_u to all other nodes n_t has to be minimal. This is again calculated by the Dijkstra's algorithm.

$$n_r = \underset{n_u \in \mathcal{N}}{\operatorname{argmin}} \left(\sum_{n_t \in \mathcal{T}'_\beta} \sum_{(n_i, n_j) \in n_u \to n_t} \mathbf{D}(i,j) \right) \tag{6.8}$$

The final traversal tree $\mathcal{T}_\beta = (\mathcal{N}, \mathcal{E})$ (Figure 6.4(d)) is created from $\mathcal{T}'_\beta$ by setting the direction of the edges in $\mathcal{E}'$ according to the expansion of breadth-first search from n_r towards the leaves.

The shape of $\mathcal{T}_\beta$ is influenced by the clustering parameter β. The granularity of clustering U_β influences the number of branches for $\mathcal{T}_\beta$ and consequently the number of cuts created. The cluster tree $\mathcal{T}_0$ for $\beta = 0$ is equivalent to $\mathcal{T}_{MST}$ because all clusters contain one frame. With increasing β trees become generally thinner with longer sequential branches. This is depicted in Figure 6.5 where the low number of clusters causes the simple shape of the tree with few cuts. The tree $\mathcal{T}_1$ for $\beta = 1$ is equivalent to purely sequential traversal because a single cluster for the whole sequence is generated. However, the spectrum of trees between MST and the sequential traversal is not completely consistent in terms of a simpler shape with increasing β. Different frame clusterings can lead to similar trees because the successive clusters can be chained together during the tree calculation and effectively produce a similar result to a coarser clustering. Hence, a large part of the β-range next to 0 produces trees which are similar to $\mathcal{T}_{MST}$. Values of β close to 1 do not result in almost sequential trees. It is not possible to create SPT by adjusting β because the cluster tree calculation stems from MST criterion.

The spectrum of possible cluster trees allows a selection of $\mathcal{T}_\beta$ which best balances the trade-off between the drift and the jumps/jitter for a given dataset. The optimal value of β has to be manually tuned according to visual evaluation of the tracked mesh sequence. Although, the approximation of the alignment error by the dissimilarity

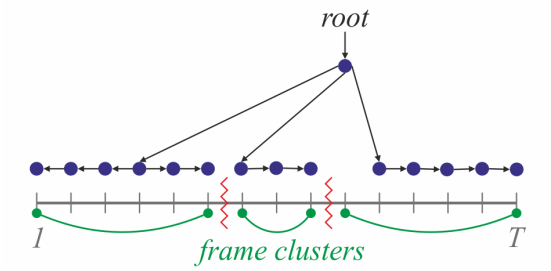


Figure 6.5: A cluster tree $\mathcal{T}_\beta$ on the input sequence clustered into three frame clusters. Notice cleaner tree structure in comparison to $\mathcal{T}$ in Figure 6.1.

can bias the tree calculation away from the optimal result, this can also be practically alleviated by adjusting β.

6.4 Non-sequential tracking using a tree

Various traversal trees $\mathcal{T}$ described can be combined with an frame-to-frame alignment technique working with the input observations $O_t = (\{I_t^c\}_{c=1}^C, M_t^g)$. The alignment technique is associated with a dissimilarity measure reflecting its capabilities. Given $\mathcal{T}$, a user manually specifies the shape and topology of the mesh $M_r = (X_r, \Gamma)$ for the root node n_r. The mesh M_r is subsequently tracked between the pairs of frames along the branches of $\mathcal{T}$ from n_r towards the leaves. The result is a temporally consistent sequence $\{M_t\}_{t=1}^T$ over the input sequence $\{O_t\}_{t=1}^T$. Multiple captured sequences of the same surface can be provided as an input and $\mathcal{T}$ spanning all of them is calculated the same way as for a single sequence.

Two different alignment techniques are used for evaluation of the non-sequential framework on various types of deformable surfaces.

6.4.1 Image-oriented frame-to-frame alignment

The frame-to-frame alignment proposed in Chapter 5 is primarily based on image information $\{I_t^c\}_{c=1}^C$. This is aimed at open surfaces captured by a narrow-baseline camera setup where the fields of view are significantly overlapping and the capture volume is relatively small. The precision of surface alignment is high but moderate motion between

frames is assumed. Therefore, this approach is suitable for facial performances and cloth deformation. The technique is used without coarse-to-fine processing extension.

6.4.2 Geometry-oriented frame-to-frame alignment

The frame-to-frame alignment proposed by Budd et al. [22] is primarily based on geometry information M_t^g. This is aimed at closed surfaces captured by a surrounding wide-baseline camera setup. The spacious capture volume allows large free-form motion of the surface between frames. The focus of this approach is robustness against fast frame-to-frame non-rigid deformations rather than high tracking accuracy. This technique is applied to whole-body performances.

The body surface is tracked non-sequentially based on MST exploiting unregistered sequence of meshes $\{M_t^g\}_{t=1}^T$ reconstructed by multi-view stereo. MST is calculated using the dissimilarity d_G described in Section 6.2. The template mesh M_r is created automatically by decimation of M_r^g at the root frame r of MST and subsequently tracked along tree branches. Between each pair of frames it is deformed according to geometrical correspondences.

An unregistered mesh M_{t-1}^g at the source frame $t-1$ is separated into surface patches of a given radius in terms of mesh topology. These rigid patches are fitted by the Iterative closest point algorithm (ICP) to M_t^g at the target frame. This provides 3D displacement vectors for the patch centres which are used as constraints in Laplacian deformation of M_{t-1}. ICP and Laplacian deformation are repeated iteratively to refine the alignment of M_t with M_t^g. Coarse-to-fine refinement is performed by increasing the number of patches between iterations to provide finer correspondence. The process stops when the individual patches contain single triangle fans.

6.5 Multi-path temporal fusion across tree branches

A drawback of non-sequential tracking based on a tree [22] is the presence of cuts between adjacent frames with different alignment paths (Figure 6.6) which can cause jumps in the final mesh sequence. To ensure smooth transitions, the original tree $\mathcal{T}$ can

be expanded with branches of a length m which extend the tracking across each cut in both directions. The expanded tree $\tilde{\mathcal{T}} = (\tilde{\mathcal{N}}, \tilde{\mathcal{E}})$ includes the tree $\mathcal{T}$ ($\mathcal{N} \subset \tilde{\mathcal{N}}, \mathcal{E} \subset \tilde{\mathcal{E}}$) as illustrated in Figure 6.6. New branches are added for every pair of adjacent frames $(t-1, t)$ where a cut is located: $(\tilde{n}_{t-1}, \tilde{n}_t) \wedge (\tilde{n}_t, \tilde{n}_{t-1}) \notin \mathcal{E}$. A chain of new nodes with interconnecting edges is created for the frames $\{t-m, ..., t-1\}$ and linked to the original node $\tilde{n}_t \in \mathcal{N}$ (similarly for the frames $\{t, ..., t-1+m\}$ and the node $\tilde{n}_{t-1}$). The new edges have weights from the respective positions in the matrix $\mathbf{D}$.

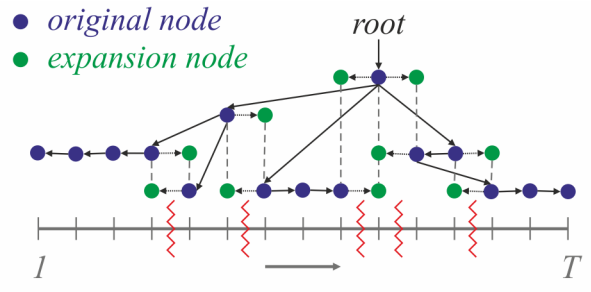


Figure 6.6: The original traversal tree $\mathcal{T}$ on the frame sequence $\{1, ..., T\}$ has blue nodes and full arrows as edges. Red cuts separate adjacent frames with different alignment paths. New nodes of the expanded tree $\tilde{\mathcal{T}}$ with $m = 1$ are marked green and new edges have dotted arrows. Vertical dashed lines join the nodes which are fused into a single result for a respective frame.

After the expansion a frame t can have multiple nodes $\tilde{n}_v$ associated with it yielding multiple solutions X_v through different alignment paths (vertical dashed lines in Figure 6.6). To combine them, every node $\tilde{n}_v \in \tilde{\mathcal{N}}$ is weighted by a coefficient η_v defined in Equation 6.9. The first term represents a tracking confidence from the root $\tilde{n}_r$ to $\tilde{n}_v$ as an inverse of accumulated dissimilarity. The second term linearly decreases the weight along the additional branches where $\tilde{n}_t$ is the last node from the original $\mathcal{T}$ on the path $\tilde{n}_r \to \tilde{n}_v$ (this term equals 1 for $\tilde{n}_v \in \mathcal{N}$).

$$\eta_v = \left(\frac{1}{\sum_{(\tilde{n}_i, \tilde{n}_j) \in \tilde{n}_r \to \tilde{n}_v} \mathbf{D}(i,j)} \right) \cdot \left(1 - \sum_{(\tilde{n}_i, \tilde{n}_j) \in \tilde{n}_t \to \tilde{n}_v} \frac{1}{m+1} \right) \tag{6.9}$$

The final vertex positions X_t for the mesh M_t are blended from all candidate positions X_v for the frame t. Equation 6.10 defines simple linear blending with the normalised coefficients η_v. According to experiments this is sufficient to produce visually pleasant

fusion across the cuts for the facial performance. An overlap of a few frames ($m = 3$) produces smooth temporally consistent mesh sequence $\{M_t\}_{t=1}^{T}$.

$$X_t = \frac{1}{\sum\limits_{\tilde{n}_v \in t} \eta_v} \sum_{\tilde{n}_v \in t} \eta_v X_v \tag{6.10}$$

The temporal fusion across cuts can be used with any type of traversal tree but it is only able to eliminate small jumps. Thus, the tree should not create large alignment inconsistencies. Also, a large number of jumps in close neighbourhood often leads to less accurate fusion which manifests as smooth local drift. From the perspective of computational overhead, more cuts introduced by a tree require more additional nodes in an expanded tree. The number of frame-to-frame alignment steps can easily exceed the original number without the fusion even for moderately fragmented traversals. According to these observations the cluster tree is favoured over MST or SPT because it generates less cuts with smaller potential inconsistencies.

6.6 Evaluation

The non-sequential approach has been evaluated for facial performance application but also for other deformable surfaces undergoing complex non-rigid motions. Table 6.1 summarises the datasets used which contain facial performances (Synthetic-skin1, Martin-skin2, DisneyFace), cloth deformation (Garment) and whole-body performances (StreetDance). The datasets DisneyFace [13] and StreetDance [101] are publicly available. All datasets provide multi-view image sequences with camera calibration and an unregistered mesh sequence (detailed description in Appendix G).

Table 6.1 specifies the configuration of experiments across the datasets. The resolution of the tracked mesh M varies between them. Also, two different frame-to-frame alignment techniques mentioned in Section 6.4 are applied according to the nature of individual datasets. The image-oriented surface alignment with the dissimilarity d_I is used for the face and cloth datasets. The geometry-oriented surface alignment with the dissimilarity d_G is used for the whole-body performance.

The image-oriented alignment has a common parameter configuration: NCC on grey-

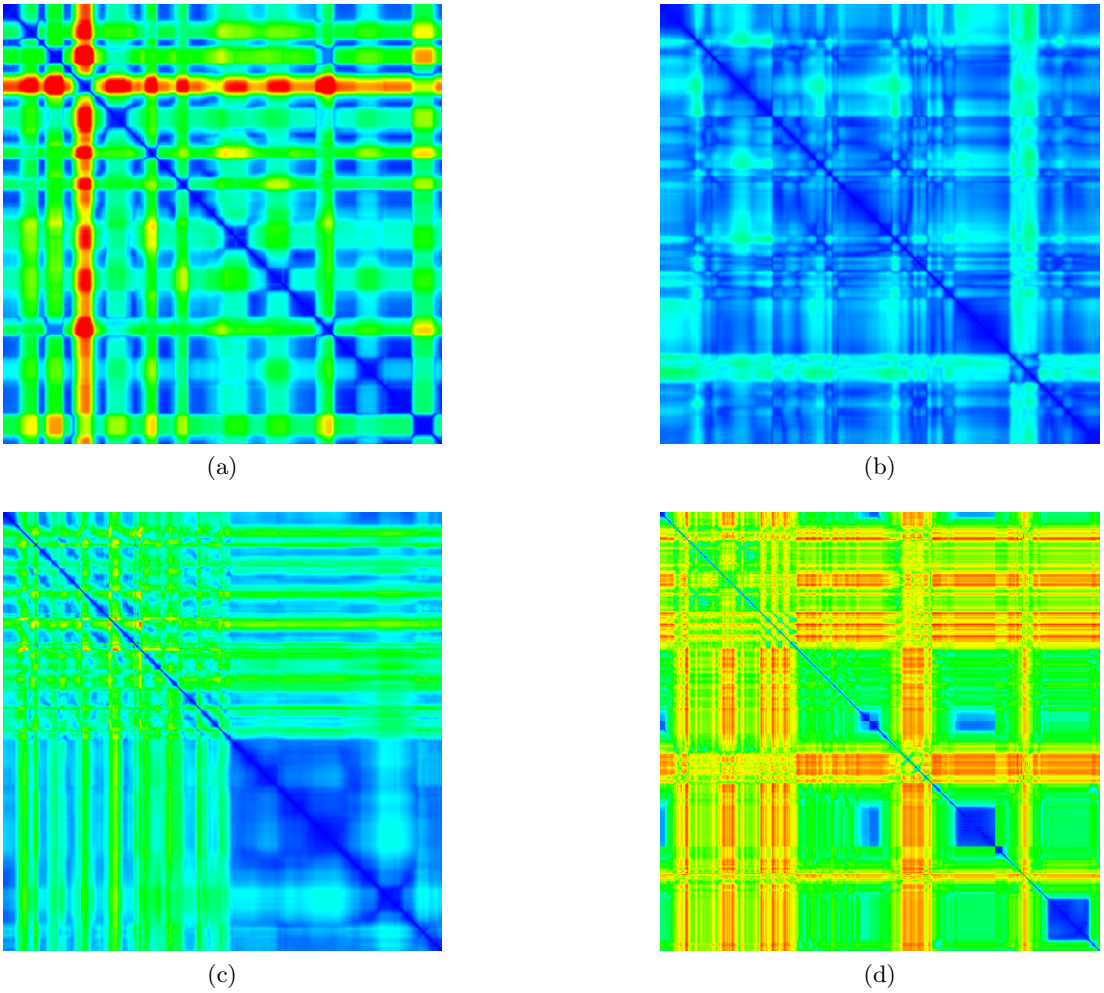
Dataset	N_v	N_f	N_v^*	$T^*(T)$	s	$q_{lim}(mm)$	$\sigma_g(mm)$
Synthetic-skin1	2689	5248	22	-	0.1	5	10
Martin-skin2	2689	5248	22	11(355)	1.0	5	10
DisneyFace	2700	5332	15	11(346)	0.5	5	5
Garment	425	768	12	9(320)	1.0	10	10
StreetDance	∼3484	∼6964	-	-	-	-	-

Table 6.1: Experimental configuration across the evaluated datasets: N_v - a number of vertices of the tracked mesh M; N_f - a number of faces of M; N_v^* - a number of surface points tracked for d_I; T^* - a number of landmarked frames for tracking of the surface points (T - total number of frames); s, q_{lim}, σ_g - parameters for robust image-oriented surface tracking. The mesh resolution for StreetDance changes slightly depending on a traversal tree because of different mesh decimation in individual root frames.

scale, $d_o = 0.2mm$, $N_o = 11$, $w_g = 1.0$, $\xi_e = 0.15$ and $\delta_e = 0.05$. The varying parameters are listed in Table 6.1. Multi-path temporal fusion across cuts is not applied unless mentioned explicitly. This is to compare directly different traversal trees without any additional enhancement step. The dissimilarity d_I is computed from N_v^* surface points tracked initially in 2D by linear predictor tracker. Training of this tracker required landmarking T^* frames out of total length T of a sequence. In the case of the dataset Synthetic-skin1 the surface points are not tracked in image sequences but their 3D trajectories are directly sampled from the ground-truth mesh sequence. The geometry-oriented alignment starts with surface patches of 7-edge radius from the central vertex. Therefore, there is 7 iterations during coarse-to-fine refinement. The dissimilarity d_G is computed using the shape histogram with 1000 bins.

Figure 6.7 visualises dissimilarity matrices constructed for the datasets based on the respective measures. They reflect the nature of a surface motion such that cross-like segments with similar colour pattern represent individual phases of the motion. The pattern along rows or columns determines how dissimilar the phase is to other movements. Blue colour marks zero dissimilarity and warmer colours increasing value.

Different traversals through the dissimilarity space $\mathbf{D}$ have been evaluated across all datasets. The following reorderings of the input sequence are compared: the standard sequential traversal ($\beta = 1$), the non-sequential traversal according to MST ($\beta = 0$), the non-sequential traversal according to SPT (no association with β) and the non-sequential traversal according to a cluster tree ($\beta = (0, 1)$). Multiple traversal trees $\mathcal{T}_\beta$



(a) (b)

(c) (d)

Figure 6.7: Dissimilarity matrices for the dataset Martin-skin2(a), DisneyFace(b), Garment(c) and StreetDance(d) (Synthetic-skin1 in Figure 6.2(a)). Colour mapping is for all datasets the same (blue = 0, red = 15) except StreetDance which uses the dissimilarity d_G (blue = 0, red = max value).

are generated for the proposed cluster-based approach to explore the spectrum of possible tree shapes between the sequential traversal and MST. The clustering parameter β is sampled in the way that distinct clusterings U_β with varying number of clusters are created for a given $\mathbf{D}$. This produces a variety of cluster trees with different structures.

Appendix C contains tables which list traversal trees evaluated for each dataset. Each tree is described by several properties: β value, the number of clusters, the number of branches, average branch length and the number of cuts. A few trends can be observed across the spectra of trees in all datasets. The number of frame clusters increases from

one cluster for $\beta = 1$ (sequential traversal) to the number of clusters equal number of frames for $\beta = 0$ (MST). SPT is not compatible with the cluster tree algorithm, therefore no clustering U_β is related to it. However, it operates directly at the level of individual frames. The spectrum of trees starts from a single branch with no cut for the sequential traversal ($\beta = 1$). The branching structure becomes more complicated introducing more cuts as β decreases. For $\beta < 0.6$ the trees are very similar to MST ($\beta = 1$) which closes the β-range. SPT is an extremely branched tree which has direct transitions from the root to all other nodes, and therefore cuts between all frames.

The temporal consistency of mesh sequences resulting from the evaluated traversals has been visually assessed from the perspective of drift versus jumps. This visual assessment takes into account the fact that sudden jumps of the mesh are more distracting for a viewer than smooth local drift. Several main observations have been made for all datasets. Sequential traversal inevitably leads to significant accumulation of errors and latter parts of the sequence have imprecise temporal consistency. Direct frame-to-frame alignment from the root frame to all other frames often fails in the case of SPT, especially during non-neutral expressions, resulting in very unstable mesh sequence. MST reduces drift in comparison to the sequential traversal significantly because difficult motions are approached by tracking from both sides in time. Thus, alignment errors occurring during these motions do not influence subsequent frames. However, there is often a large number of jumps because MST over-fragments the sequence. The best cluster tree is selected from the range $\beta = (0, 1)$ according to the tracking result (marked in tables in Appendix C). This is compared to MST which is the best previous non-sequential approach. The comparison favours the cluster tree in all datasets and discussion in following sections focuses on this point. Due to the visual nature of all results the reader is encouraged to watch the supplementary video.

SAD error on unwrapped surface textures introduced in Section 5.5.3 is used for quantitative comparison of the traversals on real-world datasets. The reference texture to compare the rest of a sequence to is taken from the root frame of the tree. Per-frame SAD error does not quantify visual severity of alignment inconsistencies across cuts. However, it still indicates accumulation of tracking imprecision and its distribution over the input sequence. A weakness of this measure is that alignment errors can be over-

ridden by appearance change during large deformation. Despite this fact, differences between different traversals are visible even at extremes of expressions.

6.6.1 Synthetic facial performance

The dataset Synthetic-skin1 is derived from the real performance Martin-skin2 to provide a ground-truth sequence with realistic facial dynamics. The dissimilarity matrices of Synthetic-skin1 and Martin-skin2 are similar in Figures 6.2(a) and 6.7(a). The cyclic structure with high-dissimilarity cross-like segments illustrates alternation between different emotive expressions and the neutral expression. The dissimilarity d_I is computed using 3D trajectories of vertices selected from the ground-truth sequence $\{M_t^{GT}\}_{t=1}^{T}$. Thus, the dissimilarity is ideal in the sense of perfect motion estimation for the surface points used.

Figure 6.8 illustrates temporal consistency achieved by the individual traversals. MST achieves temporal alignment with a few minor glitches which is superior to both the sequential and SPT traversals. The best cluster tree $T_{0.99}$ yields visually similar result to MST because the image-based frame-to-frame alignment achieves high accuracy on this sequence. Generally, there are small differences among the majority of the tested cluster trees due to the lower complexity of synthetic data.

The synthetic dataset allows quantitative evaluation of the traversals with respect to the ground truth. The ground-truth error defined earlier as the average Euclidean distance between corresponding vertices does not explicitly capture perceptual quality of tracking across the cuts. But this shows whether a tree improves globally the tracking accuracy throughout the whole sequence. The difference of the mesh sequences to the ground-truth sequence is depicted in Figure 6.9. The sequential tracking accumulates alignment errors gradually (Figure 6.9(first row)) whereas SPT has large errors during extrema of expressions (Figure 6.9(fourth row)). This is also visible in the average per-frame error plotted across the whole sequence (Figure 6.10).

The MST result has small imprecisions predominantly around the eyes and mouth and the errors do not accumulate significantly over time (Figure 6.9(second row)). The cluster tree $T_{0.99}$ has a similar error distribution across the face as MST (Figure 6.9(third

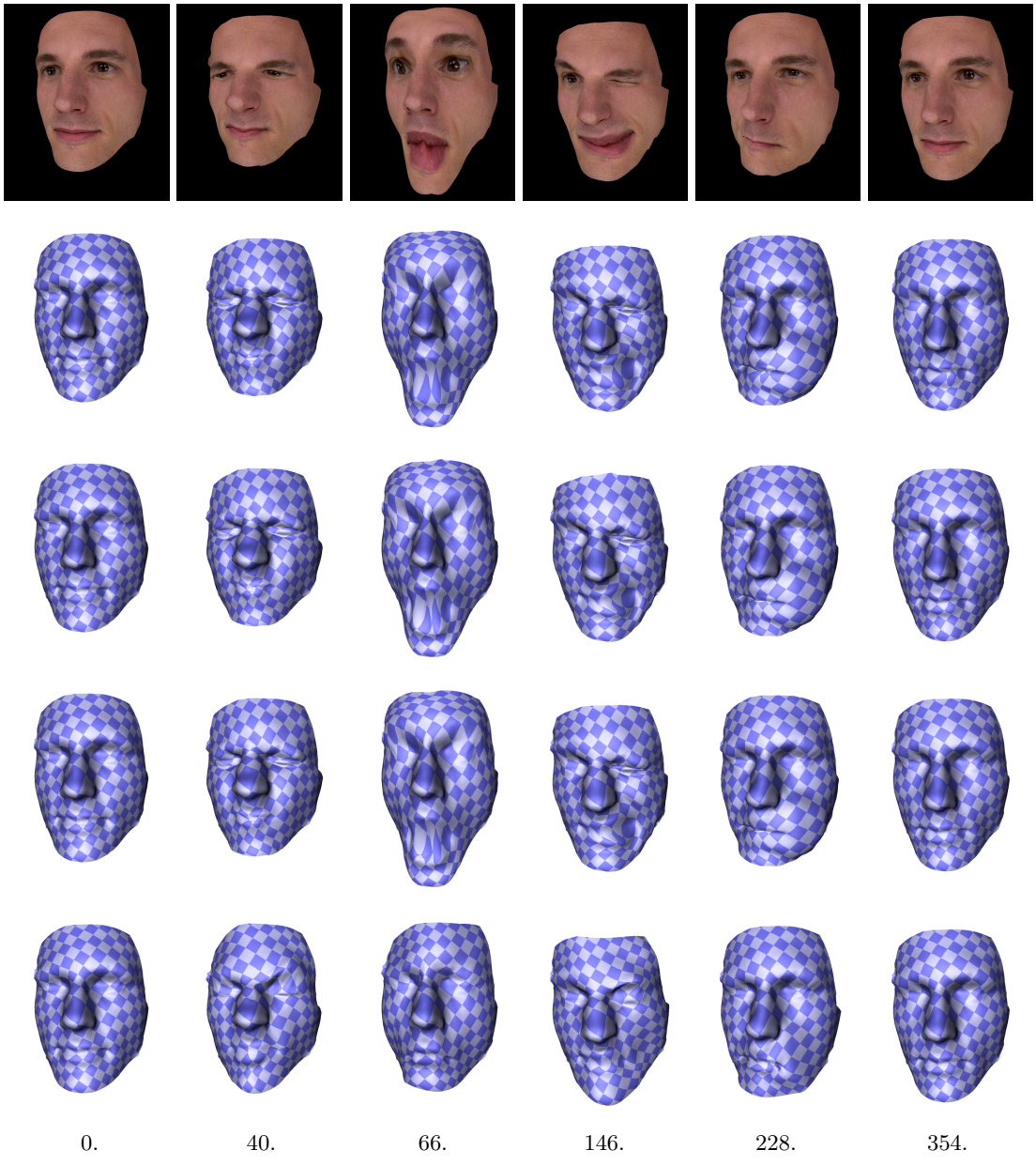


Figure 6.8: Snapshots from the temporally consistent mesh sequence for the dataset Synthetic-skin1: input images from one of the views (first row), sequential (second row), MST (third row), cluster tree (fourth row) and SPT (fifth row). The left most column represents the start/reference frame and the right most column the end frame of the sequence. Actual frame numbers are denoted.

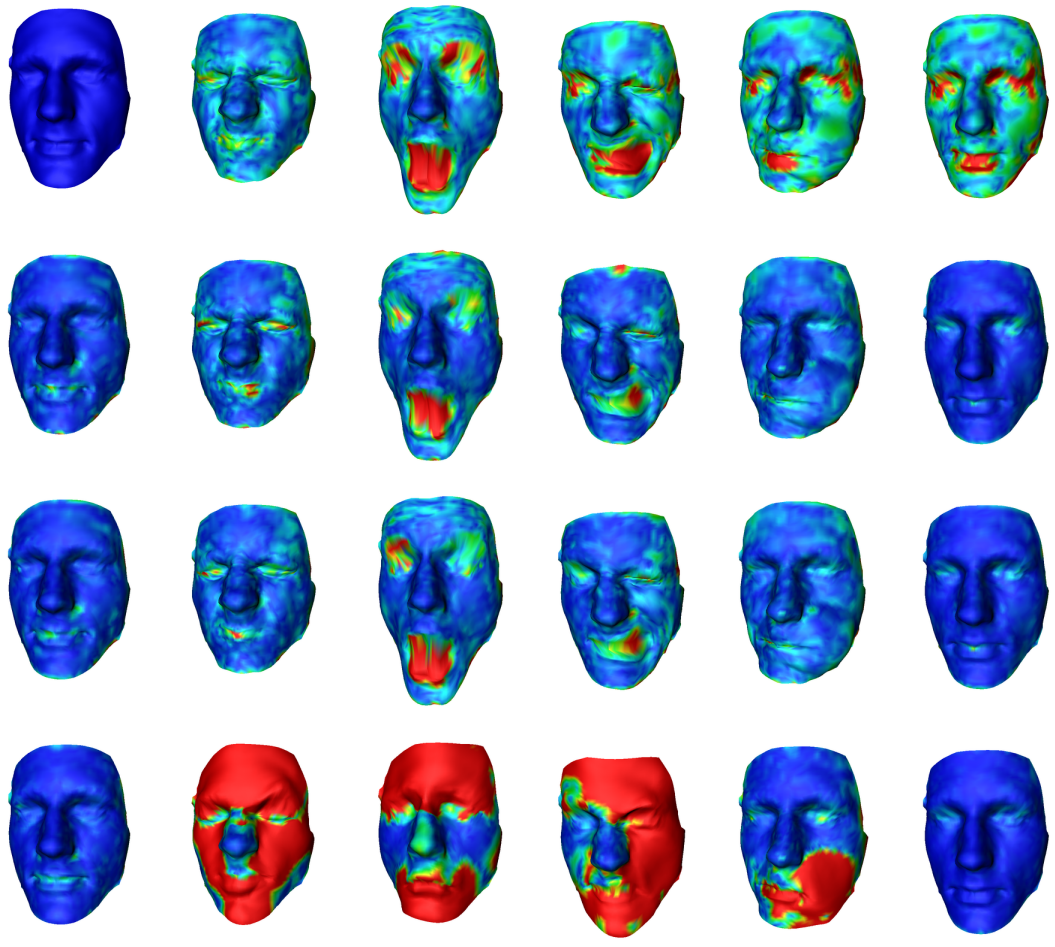


Figure 6.9: Difference of the temporally consistent mesh sequence to the ground truth for the dataset Synthetic-skin1: sequential (first row), MST (second row), cluster tree (third row) and SPT (fourth row). Euclidean distance to corresponding vertices in the ground truth is visualised across the face (blue = $0mm$, red = $2mm$).

row)). The trend of per-frame error in Figure 6.10 fluctuates for both traversals with maxima at the peaks of facial expressions. But $T_{0.99}$ yields a lower error than MST in some cases.

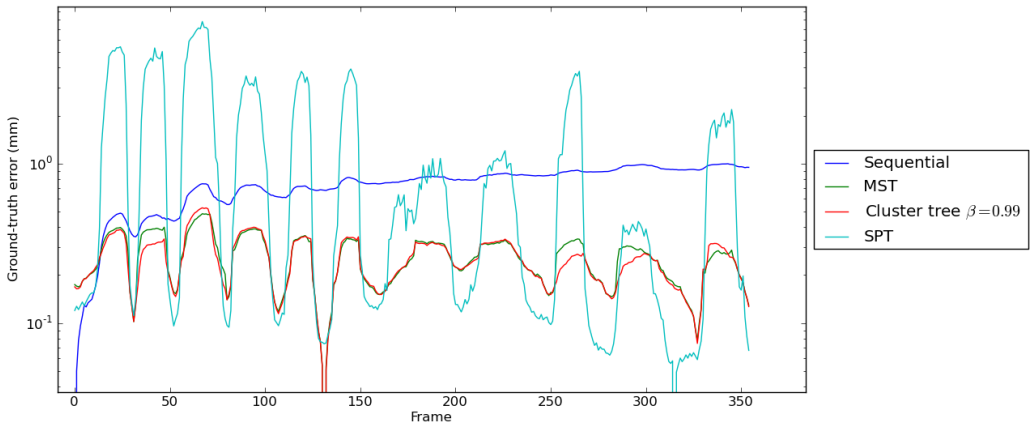


Figure 6.10: Average Euclidean distance across all vertices to the ground truth for the dataset Synthetic-skin1. The error axis has a logarithmic scale. Minima of individual curves are situated at the root frames.

Figure 6.11 plots overall ground-truth error for the whole sequence across the full spectrum of traversals evaluated (Table C.1). The sequential and SPT traversals are extremes with large inaccuracy on both ends of the spectrum. The cluster trees in between achieve much better results and the majority of them have similar average imprecision $0.25 - 0.26mm$ per vertex which shows high quality of the tracking. A few trees surpass MST, $\mathcal{T}_{0.99}$ gives the best visual result and also has the lowest vertex error: mean $= 0.250mm$, standard deviation $= 0.409mm$. Although, the ground-truth error does not quantify explicitly glitches due to the cuts, the graph in Figure 6.11 correlates well with the visual assessment of the mesh sequences.

6.6.2 Facial performance

The dataset Martin-skin2 provides a real-world performance with fast changes between various exaggerated emotions which is visible in the dissimilarity matrix in Figure 6.7(a). The matrix is computed from the same set of surface points as in Synthetic-skin1 but they are actually tracked which leads to increased noise in the dissimilarity.

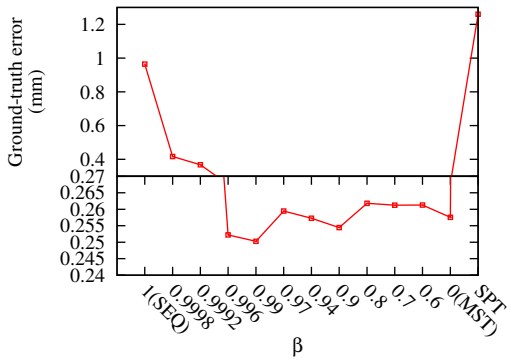


Figure 6.11: Overall ground-truth error for the dataset Synthetic-skin1 across different traversals.

Figure 6.12 compares temporally consistent mesh sequences for different traversals where the sequential traversal suffers from local drift around the eyes and lips and SPT traversal from large mesh deformations. The fragmentation in MST does not show as visible jumps in most cases because of accurate tracking in spite of weak skin texture. The qualitatively best cluster tree $\mathcal{T}_{0.95}$ improves over MST by eliminating several small glitches around the eyes and on the lips which is visible in the video. Snapshots in Figure 6.12(fourth row) illustrate accurate temporal alignment throughout the performance in spite of its challenging nature. SAD error on unwrapped facial textures in Figure 6.13 has a similar error profile for MST and the cluster tree $\mathcal{T}_{0.95}$ but $\mathcal{T}_{0.95}$ is slightly better in some emotions. Despite the real-world complexity of Martin-skin2, the quality of temporal consistency is comparable to Synthetic-skin1.

Different tracking traversals are also evaluated for the dataset DisneyFace released by Beeler et al. [13]. They are compared to the result of non-sequential tracking method by Beeler et al. which produces temporally consistent mesh sequence with high resolution (~ 1200000 vertices). This resolution is sub-sampled to $2,700$ vertices for the comparison purposes to make computational time for our processing tractable. The initial mesh at the root frame is taken from the decimated mesh sequence so that motion of the same surface points is tracked by both approaches. The performance contains natural speech with little motion outside of the mouth area. The relatively slow overall motion is visible in the corresponding dissimilarity matrix in Figure 6.7(b) which does not have a strong structure.

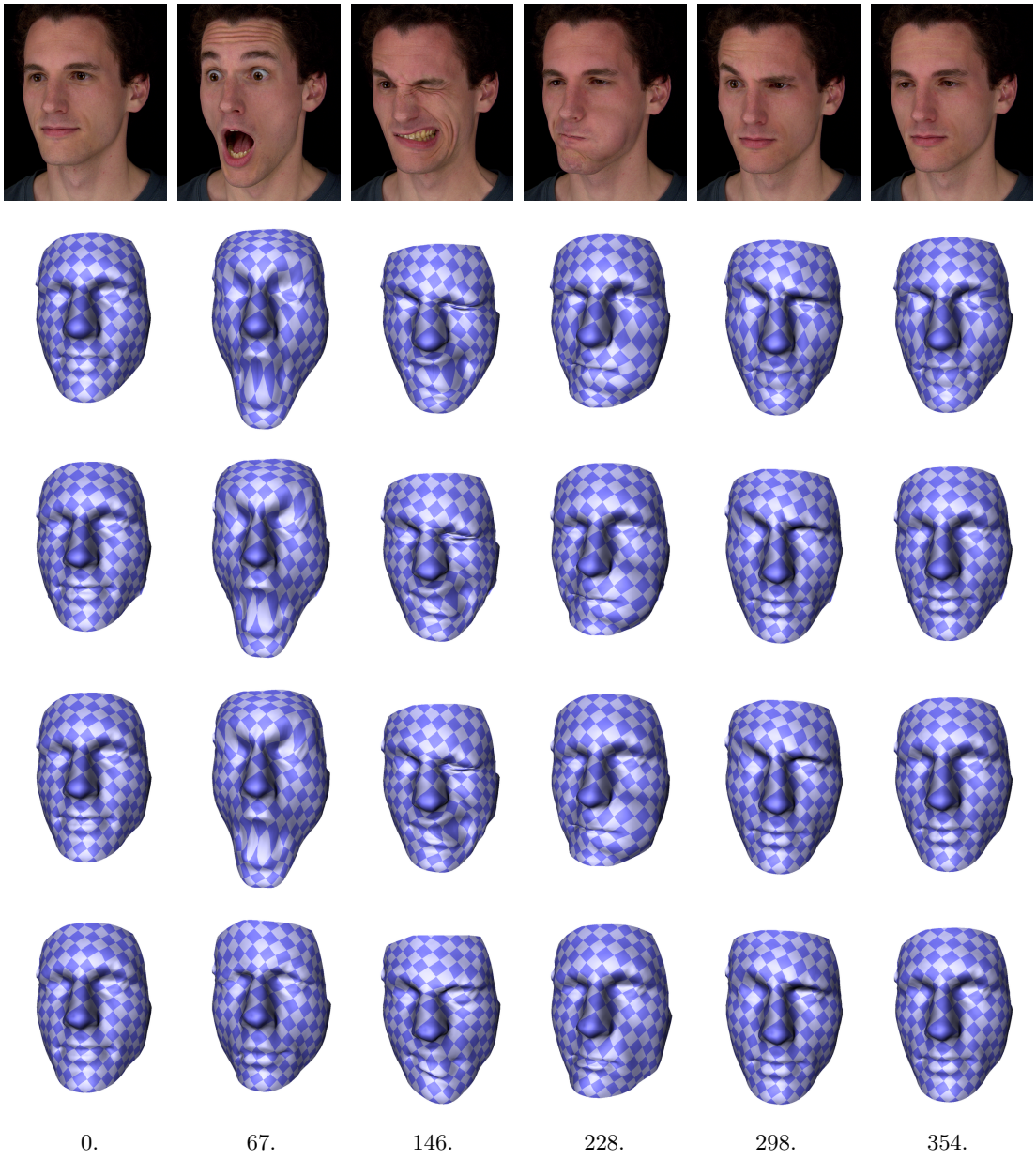


Figure 6.12: Snapshots from the temporally consistent mesh sequence for the dataset Martin-skin2: input images from one of the views (first row), sequential (second row), MST (third row), cluster tree (fourth row) and SPT (fifth row).

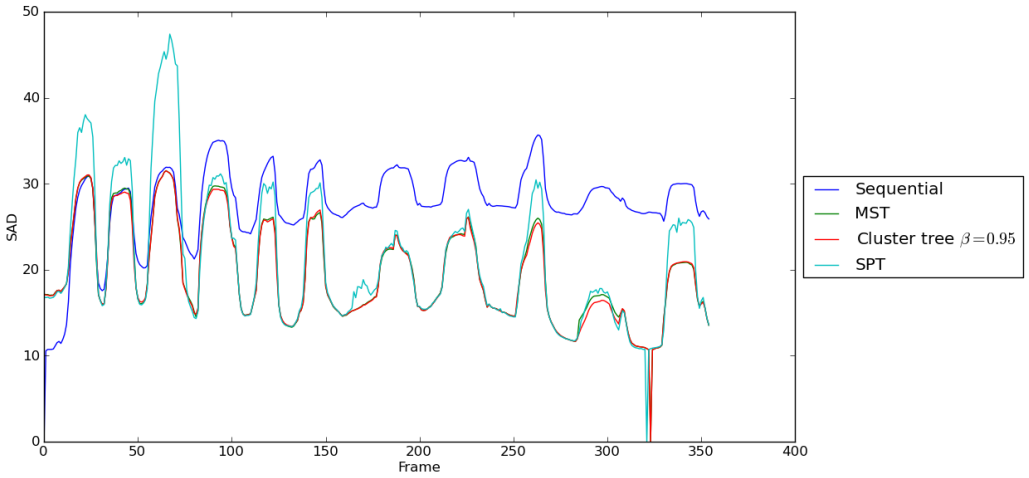


Figure 6.13: Per-pixel SAD error on unwrapped facial textures for different traversals on the dataset Martin-skin2.

Figure 6.14(second row) illustrates only accurate tracking according to the best cluster tree $\mathcal{T}_{0.98}$. The reason is that all traversals are visually similar apart from SPT which is inaccurate. Differences between them are more visible in the quantitative comparison using SAD error (Figure 6.15). The tree $\mathcal{T}_{0.98}$ achieves slightly lower error than MST over the majority of the sequence. However, the average pixel difference summed across all channels is in the range of $4-8$ colour levels which confirms high-quality temporal alignment. The sequence is not challenging enough for the robust alignment algorithm to demonstrate the benefits of the proposed cluster tree traversal.

Comparison to the result by Beeler et al. is calculated in the same way as for the ground truth on the dataset Synthetic-skin1. Note that the difference between mesh sequences may be due to the tracking errors in either approach. The per-frame average distance for the evaluated traversals in Figure 6.16 shows that the best mesh sequence by $\mathcal{T}_{0.98}$ is also the closest one to the mesh sequence by Beeler et al. The average vertex distance across all frames for $\mathcal{T}_{0.98}$ is $0.192mm$ with the standard deviation $0.225mm$. Spatial distribution of the difference is visualised across the face at the selected frames in Figure 6.14(third row). Qualitatively, the tree $\mathcal{T}_{0.98}$ yields a slightly larger drift on the inner lip than Beeler's result in some situations, otherwise they are hard to distinguish visually.

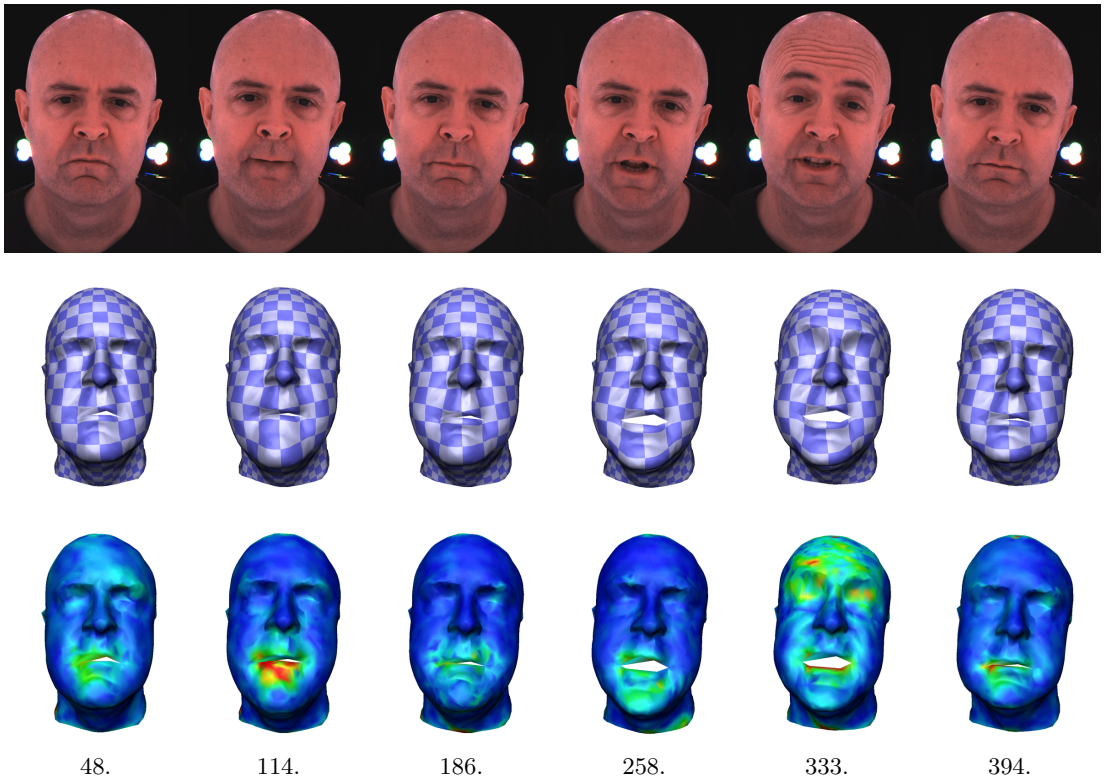


Figure 6.14: Snapshots from the temporally consistent mesh sequence for the dataset DisneyFace: input images from one of the views (first row), cluster tree (second row) and difference between cluster tree and Beeler et al. [13] (third row). The difference is visualised as the Euclidean distance of corresponding vertices across the face (blue = $0mm$, red = $2mm$).

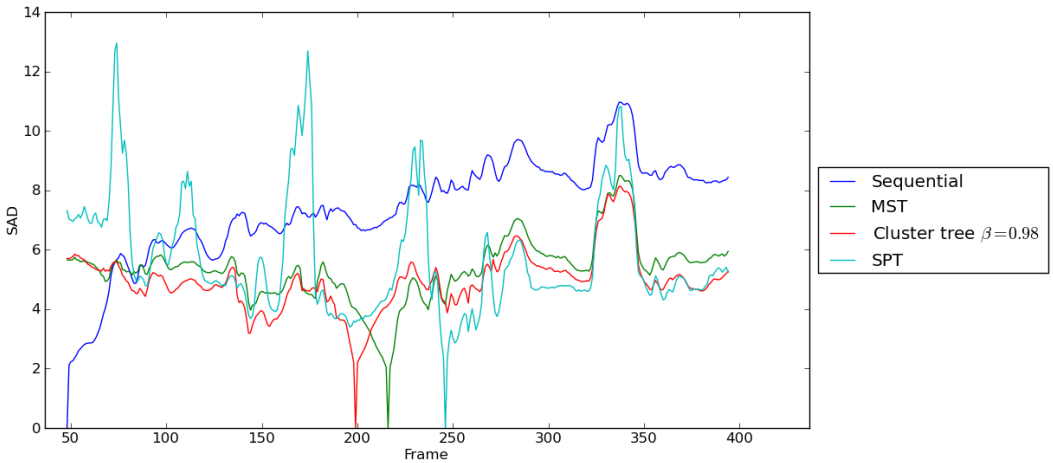


Figure 6.15: Per-pixel SAD error on unwrapped facial textures for different traversals on the dataset DisneyFace.

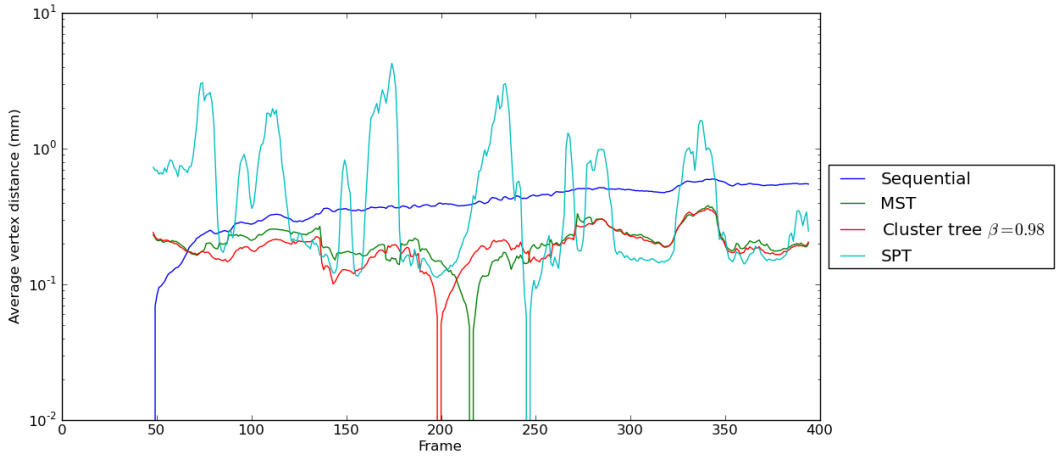


Figure 6.16: Average Euclidean distance across all vertices to the result by Beeler et al. on the dataset DisneyFace. The distance axis has a logarithmic scale. Minimal distances on individual curves are zero because initial meshes at the root frames are sampled from Beeler's mesh sequence.

6.6.3 Cloth deformation

The dataset Garment contains deforming cloth as an example of different type of surface which can be handled by the proposed tracking framework. The sequence captures motion of a loose textured top on upper torso of an actress. The top completely fills the field of view for all cameras, hence only a rectangular area in the centre is tracked (marked in Figure 6.17(first row)). The dissimilarity matrix in Figure 6.7(c) illustrates that the first half of the sequence contains fast largely repetitive motion (cyclic waving across the top) and the second half contains little non-rigid deformation (mostly translational movement during slow deep breathing).

Figure 6.17 shows example meshes from temporally consistent sequences for the different traversals where sequential and SPT traversal provide low-quality results (visible in the video). MST achieves better temporal alignment but there are many noticeable jumps due to excessive branching in repetitive motions. The cleaner structure of the best cluster tree $\mathcal{T}_{0.994}$ largely eliminates these artefacts. This is noticeable in frames 350, 388 and 451 where errors are accumulated close to the cuts in MST. SAD error on unwrapped cloth textures in Figure 6.18 has also slightly lower profile for $\mathcal{T}_{0.994}$ than MST. The improvement over MST by the cluster tree is visually bigger for the

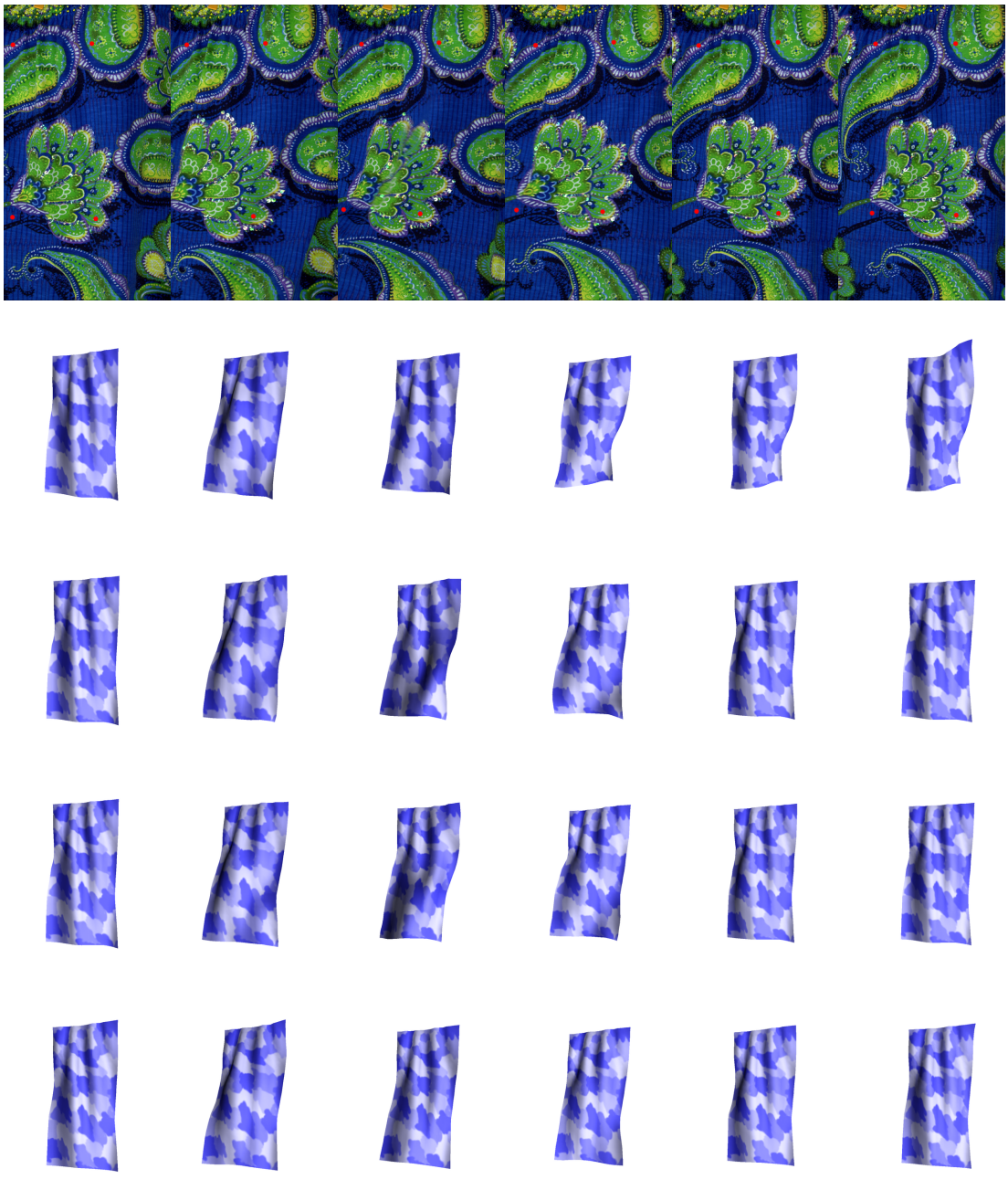


Figure 6.17: Snapshots from the temporally consistent mesh sequence for the dataset Garment: input images from one of the views (first row), sequential (second row), MST (third row), cluster tree (fourth row) and SPT (fifth row). Corners of the tracked area are marked red in the input images.

cloth than for the face datasets because of more challenging surface motion. Drift and jumps are generally more severe than for the faces because of the higher deformation complexity and motion blur which complicates the image-oriented alignment.

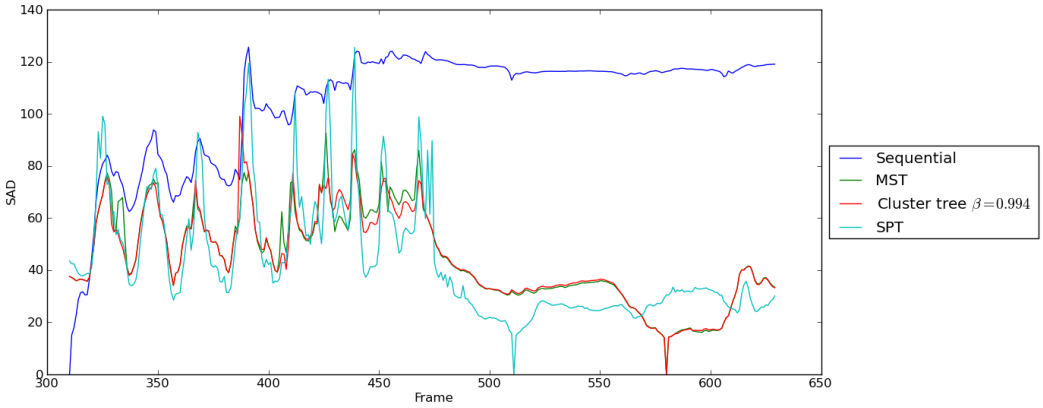


Figure 6.18: Per-pixel SAD error on unwrapped cloth textures for different traversals on the dataset Garment. Minima of individual traversals are situated at their root frames.

6.6.4 Whole-body performance

The dataset StreetDance represents another type of data where the proposed non-sequential approach is beneficial. This features whole-body performance of a break-dancer in loose uniform clothing. The sequence is composited from 3 different takes (Free, KickUp, FlashKick) to demonstrate the ability of non-sequential traversal to align the data across separate motions of the same surface. In this case, a closed mesh is tracked on unregistered mesh sequence reconstructed from 8 cameras surrounding a performer. The raw meshes capture the shape of the body, but do not contain details such as fingers or facial features. Their quality occasionally suffers from motion blur or self-occlusions during complicated break-dance moves. Moreover, the geometry-oriented frame-to-frame alignment (Section 6.4.2) is challenged by changes in surface topology such as the limbs joining the body if they are in close proximity.

The dissimilarity matrix in Figure 6.7(d) shows many fast movements spanning short frame segments which are generally quite different from each other. The matrix contains



Figure 6.19: Snapshots from the temporally consistent mesh sequence for the dataset StreetDance: input images from one of the views (first row), sequential (second row), MST (third row), cluster tree (fourth row) and SPT (fifth row). Corners of the tracked area are marked red in the input images.

more noise, especially in slow-motion parts, than in the other datasets due to less precise dissimilarity d_G. Because of over-fitting to d_G, MST is very branched with many more cuts than cluster trees (Table C.5).

Figure 6.19(second row) shows large drifts which twist and crease up the mesh for the sequential traversal. SPT results in many failures of direct alignment from the root frame (Figure 6.19(fifth row)). The MST result contains significantly less drift but over-fragmentation of slow-motion segments causes jitter of the mesh. The gross errors such as deformed limbs in frames 167, 298, 868 in Figure 6.19(third row) are due to frequent transitions in the middle of complicated movements.

The best cluster tree $\mathcal{T}_{0.996}$ enforces sequential tracking in slow-motion segments and in large parts of complicated movements. This eliminates distracting jitter and removes the majority of gross errors (e.g. frame 868 in Figure 6.19(fourth row)). The increased local drift at the peaks of complicated movements is perceptually more plausible than fast alternation between differently distorted meshes. Even the best result by $\mathcal{T}_{0.996}$ does not approach the precision of tracking in the face or cloth datasets. This is given by much more challenging surface motion and less accurate geometry-oriented alignment.

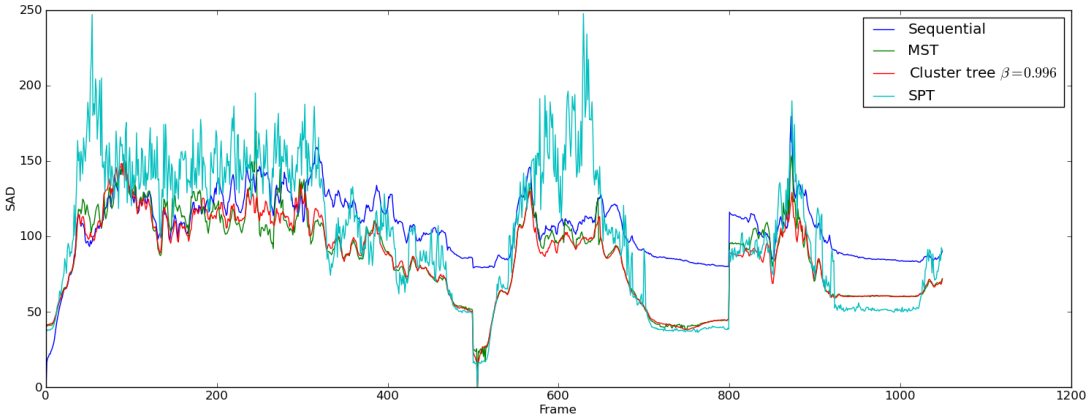


Figure 6.20: Per-pixel SAD error on unwrapped body textures for different traversals on the dataset StreetDance. Minima of individual traversals are situated at their root frames.

To compute SAD error, the surface texture is back-projected into the UV domain of the aligned meshes differently than for the previous datasets. Assignment of views for regions in the UV domain changes over time due to changes of the body pose

with respect to all cameras. Also, some mesh triangles are occasionally invisible to all cameras so they are excluded from texture comparison between frames. Figure 6.20 shows low performance by the sequential and SPT traversal. Profiles for MST and the cluster tree $\mathcal{T}_{0.996}$ favour slightly the cluster tree, mostly during complex motions.

To quantitatively judge temporal coherence of the tracked mesh in slow-motion segments, acceleration across all vertices is observed. Any fast changes in the acceleration indicate jitter or jumps in the aligned mesh sequence better than SAD error. This evaluation is applicable for the whole-body dataset because of the larger magnitude of the errors than in the other datasets. Figure 6.21 shows average acceleration across all vertices for MST and $\mathcal{T}_{0.996}$ for a segment where the dancer stands still. The peaks represent high acceleration related to the jitter of mesh. The tree $\mathcal{T}_{0.996}$ significantly reduces acceleration spikes in comparison to MST.

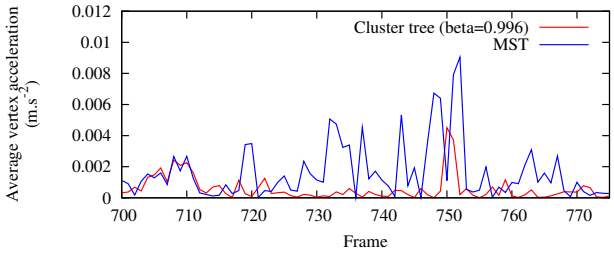


Figure 6.21: Average vertex acceleration for MST and the best cluster tree $\mathcal{T}_{0.996}$ for a segment in the dataset StreetDance where the dancer stands still. The acceleration peaks correspond to high-frequency jitter in the aligned mesh sequence.

6.6.5 Results of multi-path temporal fusion

Multi-path temporal fusion across cuts created by a traversal tree smooths out possible alignment inconsistencies. The resulting temporally consistent mesh sequence does not contain abrupt jumps or jitter, thus is visually more pleasing. The effect of fusion is shown on the best cluster trees for the face datasets Synthetic-skin1 and Martin-skin2. This also demonstrates the quality of the final results for both datasets using the complete tracking framework.

Both trees $\mathcal{T}_{0.99}$ for Synthetic-skin1 and $\mathcal{T}_{0.95}$ for Martin-skin2 introduce 19 cuts. The expansion branches across cuts have the length set to $m = 3$, therefore the both expan-

ded trees have 469 nodes in comparison to 355 nodes in the original trees. The available ground truth for Synthetic-skin1 allows quantitative comparison with/without the fusion across cuts. The comparison to the ground truth over the whole sequence in Figure 6.22 puts the result by the expansion tree $\tilde{\mathcal{T}}_{0.99}$ ahead of the original cluster tree $\mathcal{T}_{0.99}$. The average error for all frames is $0.244mm$ with standard deviation $0.405mm$ for the fusion across cuts and $0.25mm$ with standard deviation $0.409mm$ without the fusion. In the video, per-vertex error across the face is visualised as heat map. The error pattern changes abruptly for the selected cut which indicates a jump in the mesh sequence when the fusion is not applied. The coherent change of the error pattern for the fusion means smooth transition between different alignment paths.

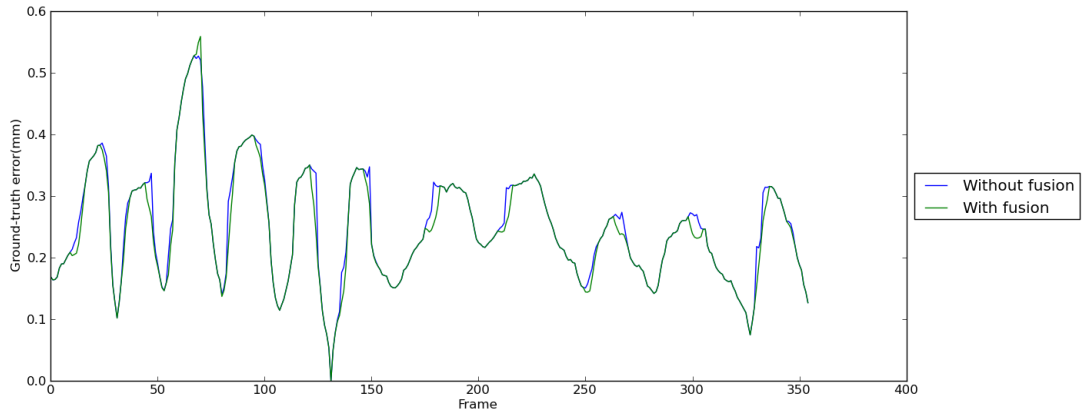


Figure 6.22: Average Euclidean distance across all vertices to the ground truth for the dataset Synthetic-skin1.

The expansion tree $\tilde{\mathcal{T}}_{0.95}$ eliminates few minor glitches introduced by $\mathcal{T}_{0.95}$ on Martin-skin2. An example of smoothing out a glitch in the eyes is presented in the video. The fixed texture on the face deforms smoothly with the fusion applied instead of a sudden jump across the cut without the fusion. Note that multi-path temporal fusion only makes the aligned meshes more coherent locally around a cut and does not globally fix the drift along alignment paths leading to a cut. Hence, if one or both alignment paths accumulate a significant error, the fusion between them replaces a jump with smooth swimming of the mesh on the actual surface. Other traversal trees (MST, SPT) benefit from fusion as well. The best results are still achieved with a cluster tree which limits the number of inconsistencies. This has an advantage in terms of

computational overhead because a smaller number of expansion nodes leads to less additional tracking.

6.6.6 Relationship between dissimilarity and alignment error

Experiments in this section empirically estimate the relationship between dissimilarity and alignment difficulty to justify the use of the measures d_I and d_G for image-oriented and geometry-oriented alignment respectively. Ideally, the dissimilarity should be compared against the actual amount of error introduced by the alignment between frames. However, this error is not available because that would require a prior correct solution of the alignment. Therefore, the error reported by the alignment method is used to approximate the true error. This is an approximation because no tracking approach knows its actual accuracy.

The error $\bar{e}_I$ for the image-oriented alignment method (Section 6.4.1) is derived from 3D patch matching error e_i. The final errors e_i after cooperative optimisation are averaged across all patches to obtain $\bar{e}_I$ for a particular frame-to-frame transition. The error $\bar{e}_G$ for the geometry-oriented alignment method (Section 6.4.2) is derived from patch 3D trajectories during ICP fitting. A rigid patch is iteratively moved from a pose at the previous frame to a target mesh in the current frame. The length of this trajectory reflects the magnitude of the actual surface motion. It is sensible to assume that larger motion is estimated with larger error. If the convergence of patch fitting is difficult, the patch trajectory is longer than the surface motion, thus indicating higher error. The error $\bar{e}_G$ for a particular frame-to-frame transition is represented by the average length of all patch trajectories.

The relationship between d_I and $\bar{e}_I$ (or d_G and $\bar{e}_G$) is observed across all traversals evaluated for a particular dataset. For each traversal each frame-to-frame transitions provides a pair of values $(d_I, \bar{e}_I)$ or $(d_G, \bar{e}_G)$. A single scatter plot of all samples across the traversals shows correlation between the measures.

Figures 6.23(a,b) contain the result for the dataset Synthetic-skin1 (the traversals from Table C.1). This dataset is an ideal case to some extent because of dissimilarity values derived from the ground-truth and no changes of surface texture over time. The

relationship between d_I and $\bar{e}_I$ has a scattered increasing trend. The profile is generally constant for low dissimilarities ($d_I < 0.4$) which means similar quality of the alignment. The range $d_I = (0.4, 3.0)$ has a linear profile, so $\bar{e}_I$ proportionally increases with d_I. Beyond $d_I = 3.0$, the samples are much more scattered and do not follow the previous linear trend. These samples are almost exclusively from SPT which chooses high-dissimilarity transitions where the alignment often fails. Outliers in the mid-range of d_I come from some non-sequential transitions in MST and cluster trees which shows that d_I sometimes incorrectly predicts the alignment error. Overall, the dissimilarity d_I and the error $\bar{e}_I$ are not fully proportional because d_I is an approximate measure of alignment difficulty.

The correlation of d_I and $\bar{e}_I$ is also shown for the real-world dataset Martin-skin2 in Figures 6.23(c,d) (the traversals from Table C.2). The scatter plot is similar to Synthetic-skin1 because the performances are effectively the same. The samples are less compact because both measures contain more imprecision. The dissimilarity d_I is influenced by noise in feature tracking. The alignment is more complicated with changes of skin appearance, so the errors $\bar{e}_I$ are generally higher.

Dissimilarity d_G for the geometry-oriented alignment is evaluated across the traversals for the dataset StreetDance (Table C.5). The monotonically increasing trend between d_G and $\bar{e}_G$ in Figures 6.23(e,f) is more scattered than for d_I and $\bar{e}_I$. This indicates that d_G is less reliable measure than d_I because some alignment errors are disproportionally high for relatively low dissimilarity values. The same conclusion is supported by the samples of SPT (blue colour) which are outliers to a lesser extent than in Figures 6.23(a,c). Their d_G values are not much higher than for precise alignments, although the errors $\bar{e}_G$ are very high.

The relationship between a dissimilarity and an alignment error is assumed to be linear for correct traversal tree calculation. Observations across different datasets show that it is non-linear in practice due to approximate nature of dissimilarity measures. However, in all cases there is a clear monotonically increasing trend between d_I and $\bar{e}_I$ (or d_G and $\bar{e}_G$) which is linear for a large part of effective dissimilarity range. This means consistent mapping between them which validates the use of d_I and d_G with their

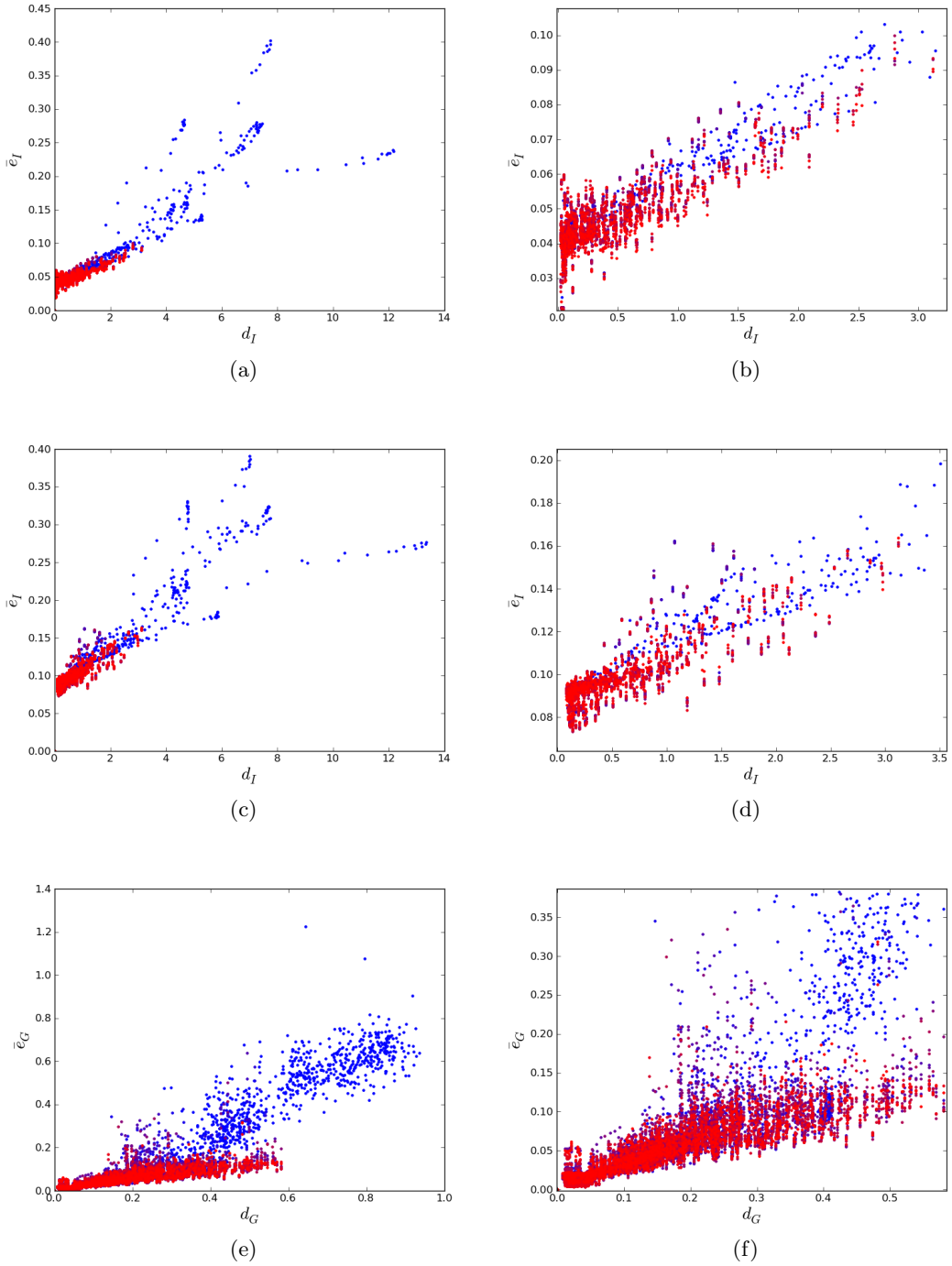


Figure 6.23: A relationship of the dissimilarity d_I and the alignment error $\bar{e}_I$ for the datasets Synthetic-skin1(a,b) and Martin-skin2(c,d). A relationship of the dissimilarity d_G and the alignment error $\bar{e}_G$ for the dataset StreetDance(e,f). The subfigures (b,d,f) contain magnified areas from (a,c,e). Colour scheme marks data samples in this order: the sequential traversal (red), the cluster trees $\beta = 1 \rightarrow 0$, MST, SPT (blue).

respective alignment methods. If this mapping is known beforehand, the dissimilarity can be transformed to a more accurate estimate of the actual alignment difficulty and improve traversal tree computation.

6.6.7 Towards optimal traversal tree

This section discusses a theoretical formulation of the optimal traversal tree in terms of perceptual quality of the resulting aligned mesh sequence. The experiments presented above empirically evaluate the quality of results for different traversals through input sequences. However, the quality of temporal alignment could be estimated from the tree before the actual tracking. The estimation process needs to assess potential alignment artefacts such as drift and jumps which can occur during the tracking. This is based on the structure of the tree and dissimilarities in the edges. Several characteristics of the traversal tree are investigated to estimate different properties of the temporally consistent mesh sequence. All these characteristics are subject to the non-linear relationship between the dissimilarity and the alignment error.

The first characteristic describes the total amount of potential errors which can occur in all frame-to-frame alignments according to the traversal tree $\mathcal{T}$. This is derived from the optimisation criterion for MST (Equation 6.1), hence the measure SEW is the sum of weights of all edges in $\mathcal{T}$ (Equation 6.11).

$$SEW = \sum_{(n_i,n_j)\in\mathcal{T}} \mathbf{D}(i,j) \tag{6.11}$$

The second characteristic is defined by the measure SPL which estimates the amount of potential drift accumulated in individual frames. The amount of drift at the frame t is related to the dissimilarity accumulated along the path $n_r \rightarrow n_t$ in $\mathcal{T}$. This is derived from the optimisation criterion of SPT (Equation 6.2), hence SPL is the sum of path lengths from n_r to all other nodes (Equation 6.12).

$$SPL = \sum_{n_t\in\mathcal{T}} \sum_{(n_i,n_j)\in n_r\rightarrow n_t} \mathbf{D}(i,j) \tag{6.12}$$

Both measures SEW and SPL do not have a notion of temporal order of frames (the same as the trees MST and SPT), therefore they do not explicitly reflect the presence of cuts introduced by $\mathcal{T}$.

The third characteristic described by the measure CUT represents the amount and magnitude of potential alignment inconsistencies at the cuts created by $\mathcal{T}$. The difference in drift accumulation between adjacent frames $t-1$ and t is related to the dissimilarity accumulated along their individual paths $n_r \rightarrow n_{t-1}$, $n_r \rightarrow n_t$ in $\mathcal{T}$. The extent of different error accumulation is defined by the length of non-overlapping parts of both paths $(n_u \rightarrow n_{t-1}) \subset (n_r \rightarrow n_{t-1}), (n_u \rightarrow n_t) \subset (n_r \rightarrow n_t)$ where n_u is a branching node which the paths separate at. This is evaluated for all pairs of adjacent frames which are not linked directly by an edge in $\mathcal{E}$: $\bar{\mathcal{E}} = \{(n_{t-1}, n_t) | \forall (n_{t-1}, n_t) \notin \mathcal{E}\}$. Equation 6.13 for CUT defines the total sum of non-overlapping sub-paths for all cuts created by $\mathcal{T}$.

$$CUT = \sum_{(n_{t-1},n_t)\in\bar{\mathcal{E}}} \left(\sum_{(n_i,n_j)\in n_u \rightarrow n_{t-1}} \mathbf{D}(i,j) + \sum_{(n_i,n_j)\in n_u \rightarrow n_t} \mathbf{D}(i,j) \right) \tag{6.13}$$

All three measures are calculated across all traversals evaluated for individual datasets in the previous experiments. Graphs discussed in following text are available in Appendix D. Figure D.1 shows profiles of SEW measure across the datasets. The general trend is a decline from the sequential traversal across cluster trees to MST which is always the minimum. The cluster trees become more sub-optimal for SEW with increasing β because more sequential ordering includes transitions with higher dissimilarity. SPT has the maximal value because direct edges to all frames contain a large amount of high dissimilarities. It is difficult to relate SEW as a theoretical estimate of the total error in a temporally aligned mesh sequence with visual assessment of the sequence. The viewer is more susceptible to the distribution of errors over time rather than their total magnitude.

SPL measure across all datasets is plotted in Figure D.2 where SPT is always the minimum. The maximum is the sequential approach which suffers the worst accumulation of errors across the whole sequence. SPL generally decreases with β towards MST.

However, some cluster trees have better characteristic than MST for some datasets. A similar general trend is observed for drift in the tracked mesh sequences in the empirical results across the datasets.

CUT measure is visualised in Figure D.3 where SPT with the cuts between all adjacent frames represents the maximum for all datasets. The sequential traversal with no cuts is the zero-value minimum. Overall, CUT decreases from MST throughout cluster trees towards the sequential traversal as the number of cuts created decreases. There are some fluctuations because a smaller number of cuts with big inconsistency can achieve similar or higher CUT value than larger number of cuts with small inconsistency. Visual assessment of jumps in the empirical results concludes a similar general trend.

Individual measures reflect certain aspects of the tracking results given a traversal tree, and therefore favour one of the specialised cases - sequential, MST or SPT traversal. A combined measure should point at the optimal traversal tree which yields the best temporally aligned mesh sequence. The opposing trends of SPL and CUT measures across the spectrum of trees illustrate the trade-off between error accumulation and cuts. This correlates with the empirical analysis where the best result balancing the drift and jumps has been obtained using a cluster tree from the middle of the tree spectrum. A combination of SPL and CUT measures should create a valley-like profile across the tree spectrum where SPT and sequential traversals have high values and the bottom of the valley lies among cluster trees. However, this is not the case across the datasets because SPT is often favoured as the minimum by the combined measures. Also, the cluster trees are not ranked in an order which correlates well with visual assessment of the tracking results. Firstly, this could be caused by unsuitable formulation of the measures. Secondly, the non-linear relationship between dissimilarity and alignment error can bias the measures away from the actual tracking quality. The relationship is different for each type of the surface, dissimilarity measure and alignment method which could be a reason for inconsistent observations across the datasets.

Definition of a single measure assessing the trade-off between potential drift and jumps consistently across different datasets is necessary. This would allow to formalise optimality of the traversal tree in terms of the resulting quality of temporal alignment.

The clustering parameter β could be selected automatically before the surface tracking instead of tuning through experimentation. Alternatively, a new algorithm for tree calculation could be designed with this measure as an optimisation criterion which would provide the optimal tree directly. Automatic calculation of the optimal traversal tree with respect to drift and jumps remains an open problem.

6.7 Conclusion

This chapter has introduced a non-sequential approach to dense surface tracking which is one of the first non-sequential methods to address the drift problem for facial performances. The order in which the tracking progresses through the input sequence is given by a traversal tree. The tree is based on a fast, approximate dissimilarity measure which estimates the difficulty of alignment between pairs of frames. Several methods of calculating the traversal tree in the dissimilarity space have been analysed. The minimum spanning tree and the shortest path tree used in whole-body tracking [52, 22] tend to over-fragment the input sequence as there is no penalty for introducing cuts.

A novel cluster tree algorithm has been proposed to reduce jumps in temporal alignment caused by a large number of cuts. The temporal order of frames is taken into account and the sequence is clustered into segments of similar frames based on low mutual dissimilarity. Sequential tracking is enforced in these segments which reduces the number of cuts. The algorithm allows calculation of a spectrum of trees between fully sequential traversal and the minimum spanning tree. Thus, trade-off between sequential drift along tree branches and non-sequential jumps across cuts can be balanced. To eliminate potential jumps, multi-path temporal fusion across cuts is introduced for any kind of traversal tree.

The proposed non-sequential surface tracking has a generic framework which can be combined with any frame-to-frame alignment method and associated dissimilarity measure. For facial performances, the robust alignment method from Chapter 5 is used for tracking along branches of a traversal tree. Dissimilarity necessary for tree calculation is derived from motion of a sparse set of surface points. The traversal tree can be calcu-

lated over multiple sequences which allows global alignment of multiple performances by the same actor.

Evaluation was conducted on facial performances, cloth deformation and whole-body performances to demonstrate generality of the tracking framework. Different traversals of the input data are compared in terms of visual quality of the temporally consistent mesh sequence: sequential, minimum spanning tree, shortest path tree and cluster tree traversal. For all datasets the cluster tree achieves visually the best results with reduced drift and the limited amount of jitter. Quantitative analysis based on stability of textures back-projected on the mesh sequence over time supports the qualitative superiority of alignment based on the cluster trees.

For facial performances, differences between the minimum spanning tree and cluster tree are relatively small on both synthetic and real data due to the high accuracy of the image-oriented alignment (Chapter 5). However, the improvement is significant for VFX applications because of the high-quality temporal consistency required. A comparison to a state-of-the-art non-sequential method for facial performances [13] shows comparable performance. Tracking of cloth deformation and whole-body performances benefits more from the cluster tree approach than facial performances. This is due to more difficult movements which are challenging for frame-to-frame alignment techniques. For whole-body performances, geometry-oriented frame-to-frame alignment is combined with the dissimilarity based on shape histograms [22].

The relationship between dissimilarity and alignment error was empirically analysed for both image and geometry-oriented methods. In both cases, it is partially non-linear because the approximate dissimilarity does not reflect exactly capabilities of the alignment technique. This biases the cluster tree calculation which assumes a linear relationship. Automatic selection of the optimal cluster tree is also complicated by this fact, thus balancing of drift and jumps in the tracked mesh sequence is done through experimentation with a cluster tree structure. Direct calculation of the optimal cluster tree consistently across different datasets remains an open problem.

In the next chapter, the non-sequential tracking of medium-resolution facial geometry is combined with high-resolution detail capture from Chapter 3.

Chapter 7

Facial performance capture

In the previous chapters, the focus has been on key aspects of 3D capture of a facial performance. Chapter 3 deals with reconstruction of skin geometric detail. Chapters 4, 5, 6 propose and evaluate approaches to achieve temporal consistency of medium-scale facial shape. This chapter describes integration of these techniques and additional processing blocks to form a complete system for facial performance capture. Design of the system reflects advantages and drawbacks of the previous work in this area.

Facial performance capture is often treated as a sequence of static reconstructions which is useful for replay purposes only. Some current systems provide per-frame normal maps with fine geometric details in real-time using PSCL [110]. However, 3D models created by integration of the normals maps are deformed due to low-frequency bias in the normals. A better solution is the combination of a medium-scale shape reconstructed using multi-view stereo with the detailed normal maps estimated using photometric stereo. Fyffe et al. [39] use photometric stereo with gradient illumination which requires high-speed cameras and the Light Stage.

These techniques lack temporal consistency of 3D models across frames which is required for further manipulation and editing of the captured data in VFX production. Furukawa and Ponce [36, 37] perform 3D tracking of a facial mesh sequentially on multi-view image sequences. Use of reference appearance from the first frame alleviates the drift problem but a dense random pattern painted on the face is required to maintain

stability of the method. Flow-based motion capture [19] deforms the template mesh using 2D optic flows calculated on the image sequences. The motion of the vertices between frames is optimised to conform to the flow fields and unregistered meshes computed by multi-view stereo at every frame. The drift due to sequential concatenation of frame-to-frame flows is partially corrected by additional optic flow in the UV domain of the mesh after the initial deformation.

Previous methods with temporal alignment do not recover skin details because resolution of the facial shape is limited by multi-view stereo capabilities. Wilson et al. [119] combine surface tracking based on 2D optic flow with Light Stage technology. Rich geometric detail in the normal maps obtained by photometric stereo is exploited to improve the optic flow computation between fully lit tracking frames. Temporal upsampling enables interpolation of the facial model for the frame with different illumination pattern necessary for the photometric stereo. The meshes and normal maps are merged into a high-resolution mesh sequence which is temporally aligned but suffers substantial drift over time. The sequential tracking across different approaches is generally unreliable over long and complex performances because of the accumulation of alignment errors due to weak skin texture. Beeler et al. [13] tackle this problem by processing the sequence in a non-sequential order using anchor frames. The tracking is performed on raw meshes from multi-view stereo and 2D optic flows similarly to [19]. Moreover, the geometric detail is approximated from skin appearance under white diffuse illumination based on an assumption that darker colour indicates a cavity. The temporal alignment is calculated for all vertices of the high-resolution mesh, thus the facial shape is represented up to the finest scale in temporally consistent mesh sequence.

This chapter presents individual building blocks and a processing pipeline of the novel capture system which targets the temporal consistency and the high fidelity of the 3D model of facial performance. Firstly, technical parameters of the capture setup and data acquisition are described. Secondly, per-frame 3D reconstruction of a face from stereo camera pairs is explained as a prerequisite for the surface tracking. Thirdly, combination of the non-sequential tracking framework based on a cluster tree and the detail capture using PSCL is presented. Furthermore, the temporally aligned mesh sequence allows correction of artefacts occurring in the normal maps. The resulting 3D

model consisting of the mesh and a UV normal map sequence is qualitatively assessed for example performances. The proposed system is also compared to the state-of-the-art [13] on a publicly available dataset.

7.1 System overview

The objective of this work has been to develop a practical 3D capture system for facial performance. The output of this system should be a sequence of high-detail 3D models capturing the facial performance of an actor. The proposed design has followed the listed requirements to overcome some limitations of existing methods.

- Accurate reconstruction of facial shape up to fine skin structure such as small wrinkles, pores, etc.
- Reconstruction of a full 3D model at every frame captured.
- Temporally consistent geometric models across the performance.
- Practical capture setup operating at standard video frame-rate and without structured or time-varying illumination.
- Model-free processing without prior assumption that a human face is captured.
- Layered representation of the geometry consisting of a medium-resolution 3D mesh and a high-resolution 2D normal map.
- Ear-to-ear coverage of a face focusing on skin areas.

The pipeline of the proposed system is illustrated in Figure 7.1. An actor is captured by two stereo camera pairs which are synchronised and fully calibrated with respect to WCS. The actor's face is illuminated by red, green and blue light from different directions.

3D reconstruction (Chapter 7): The actor's face is firstly reconstructed at every frame using stereo matching in both camera pairs. Disparity maps are obtained by a graph cut and filtered from outliers according to matching scores. Both maps for each side of the face are merged into a single mesh by Poisson surface reconstruction. The output is a temporally unaligned sequence of meshes which constrains the subsequent surface tracking.

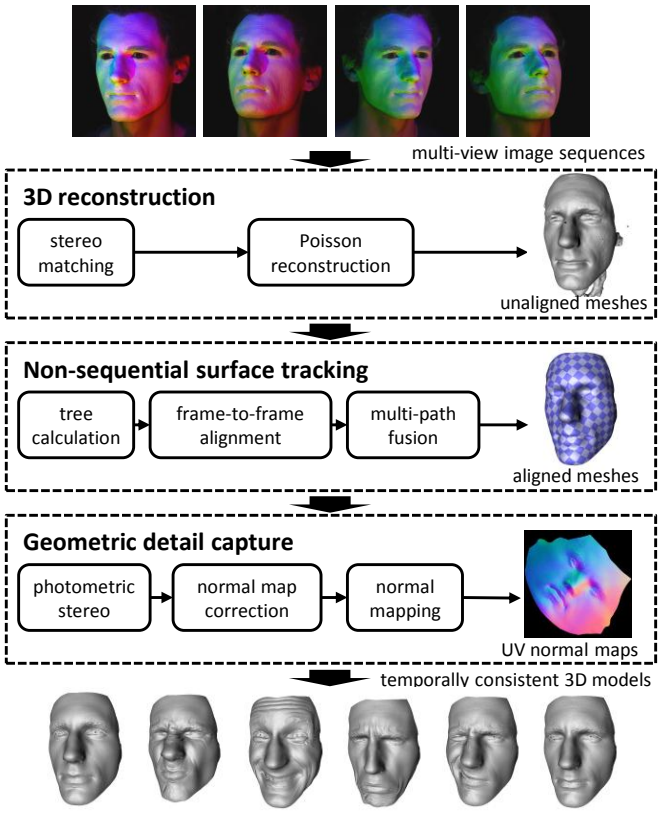


Figure 7.1: A diagram of the processing pipeline.

Non-sequential surface tracking (Chapter 6): At first, a sparse set of facial features is sequentially tracked in the 3D space throughout the performance. Pair-wise dissimilarity between all frames is established according to the spatial configuration of the points. Non-sequential traversal of the sequence is calculated using a cluster tree based on the dissimilarity among frames. The user needs to design a facial mesh for the root frame of the traversal tree which is then tracked along tree branches to all frames. Tracking is performed using a surface model of textured 3D patches associated with all mesh vertices. Frame-to-frame alignment produces an initial raw motion field for the vertices by 3D matching of the deformable patches to images and the unaligned geometry in the next frame. Motion of the mesh is then regularised with a weighted Laplacian deformation. Inconsistencies between meshes in adjacent frames aligned along different tree branches are resolved by multi-path temporal fusion. The output is a temporally aligned mesh sequence with a fixed topology.

Geometric detail capture (Chapter 3): Skin detail is reconstructed by PSCL with the aid of white uniform make-up on the face. Normal maps of the skin are obtained at every frame for one view in each stereo pair. Shadow artefacts and low-frequency bias present in the original normals are corrected exploiting the aligned mesh sequence. The corrected normal maps for each side of the face are combined into a single normal map by back-projecting onto the mesh. The temporally aligned meshes are textured by a time-varying normal map sequence stored in a fixed UV space.

The output of the pipeline is a temporally consistent 4D model of the performance capturing subtle dynamics of the face such as skin wrinkling, pore stretching, etc. The resulting accuracy of the model depends on the resolution of the base mesh and the normal map. Both resolutions can be scaled up to achieve high accuracy at the expense of longer computation time and larger data size. Note that it is possible to create a common model across multiple performances of the same actor.

7.2 Capture setup

An actor's performance is recorded by four Grass Valley Viper cameras in HD-SDI uncompressed $4:4:4$ format for high colour fidelity (Figure 7.2). Video streams are captured at $25fps$ in uninterlaced HD resolution (1920×1080 pixels). All cameras are synchronised by gen-lock signal. Gamma correction is switched off to preserve a linear response of R, G, B CCD sensors assumed by PSCL. Colour balancing is performed across all cameras to achieve similar colour rendition. This is beneficial for matching image information between the views or calculation of consistent normal maps across the views. Colour channels of all cameras are balanced by analysing sensor responses to the grey-scale step pattern and the Macbeth colour checker on a waveform monitor. The cameras are arranged into two vertical stereo pairs with a narrow baseline of $25cm$ to cover both sides of the actor's face (Figure 7.3). The distance between the centre of the capture volume and camera plane is $105cm$.

Directional colour illumination necessary for PSCL is provided by three Optikinetics Solar 250 light projectors. They are placed in a circle with a radius of $47cm$ on the same plane as the cameras. The projectors point towards the centre of the capture volume

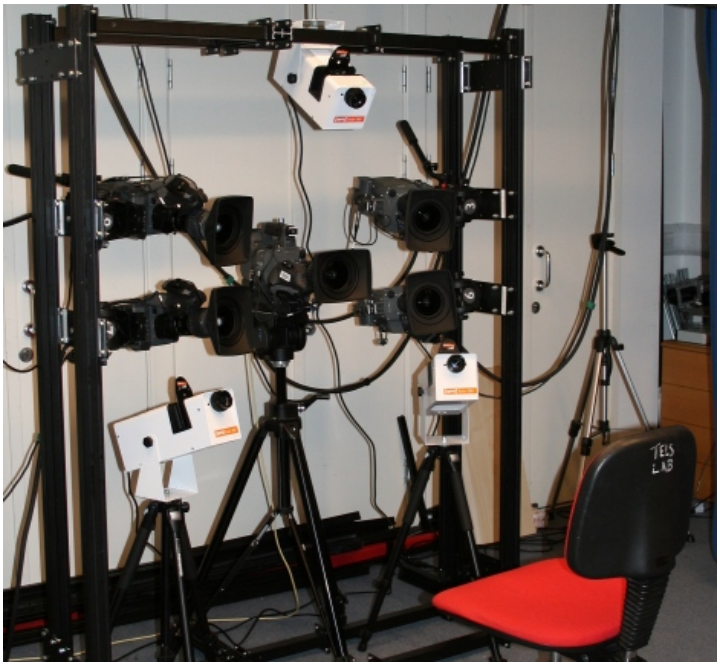


Figure 7.2: The capture setup for facial performance consists of four HD film cameras in two vertical pairs and three light projectors with colour filters. The central camera is auxiliary and is not used in the processing.

with a slant angle 24° as illustrated in Figure 7.3. The location of lights is a compromise between spatial limitations of the whole setup and the orthogonal configuration of light directions which is optimal for the photometric stereo. Therefore, they are not equidistantly spaced on the circle. Also, the small slant of the light directions reduces the size of shadows on the face which complicate calculation of surface normals. The illumination is colour-filtered to spectrally differentiate the light sources (red, green, blue light). The light projectors are extended with dichroic colour filters because of their high transmittance and high working temperature. The spectral transmittance of the filters approximately matches the spectral sensitivity curves of the R, G, B CCD sensors. Thus, each light source is almost exclusively captured by the respective sensor which minimises sensitivity of PSCL to the noise and the calibration errors.

The facial performances used in the previous chapters has been recorded under white diffuse illumination due to the focus on surface tracking. In this scenario, the actor is lit by four soft boxes placed around the cameras. The cameras record in HD-SDI uncompressed $4:2:2$ because of lower requirements on colour accuracy.

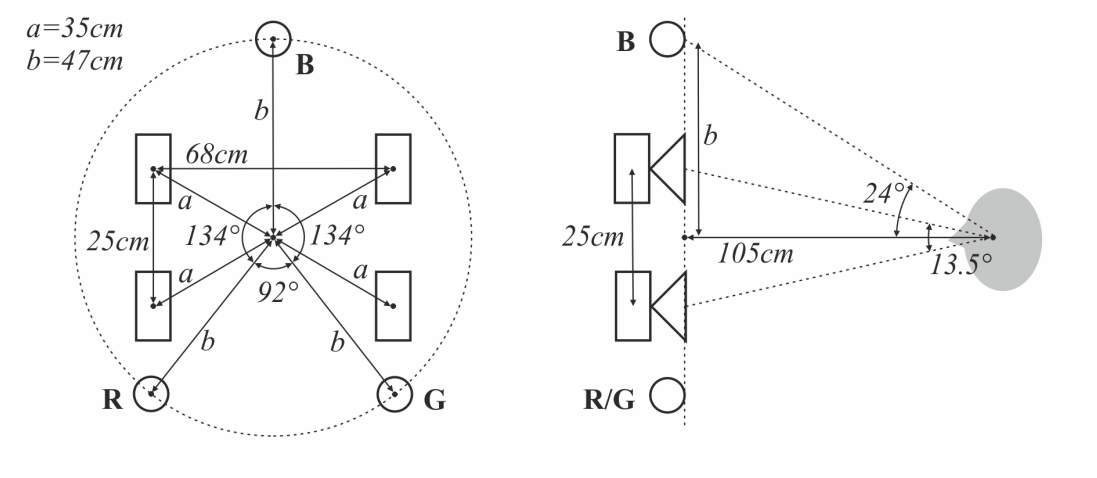


Figure 7.3: A scheme of the capture setup from the frontal and side view. The cameras are illustrated as rectangles and the red, green and blue lights as circles.

All cameras are fully calibrated with respect to the WCS anchored in the centre of capture volume. The camera model used is projective and models first order radial distortion (one radial coefficient). The principal point is fixed in the middle of the image and the pixel aspect ratio assumed to be one, thus focal lengths along rows and columns of the sensor are the same. These assumptions hold well for the Viper film camera which has highly precise build. Intrinsic parameters of individual cameras are initialised using Zhang's method [128] on images of checker-board pattern at different orientations. The radial distortion coefficient is not involved at this stage.

Initial values of intrinsic parameters are passed to wand-based multiple camera calibration [74]. Instead of sampling the capture volume with the actual wand with two colour markers, corners of the checker-board pattern across different orientations are used as 3D point samples. Intrinsic and extrinsic parameters of all cameras and the 3D point cloud are optimised together by a bundle adjustment approach. This minimises a re-projection error of all 3D points in every view which is typically around 0.25 pixels. The distortion coefficient estimated at this stage is a part of camera model, however the image sequences of a performance are undistorted before any processing. Thus, the camera model used during the processing is simplified to the basic pinhole model to accelerate operations with the camera. Calibration of the capture setup necessary for PSCL is described in Section 3.3.

7.3 3D reconstruction

An actor's performance is captured in several image sequences $\{\{I_t^c\}_{c=1}^C\}_{t=1}^T$ from different viewpoints. A shape of the face M_t^g is reconstructed at each frame t from the set of images $\{I_t^c\}_{c=1}^C$. The mesh sequence $\{M_t^g\}_{t=1}^T$ is temporally unaligned, thus there is a varying number of vertices and a varying mesh topology over time.

At first, a face region is segmented in all sequences $\{\{I_t^c\}_{c=1}^C\}_{t=1}^T$ using keying in Nuke [103]. The segmentation is based on a skin colour model which is created from several sample regions selected by a user . To include non-skin areas of the face such as the eyes and lips a morphological closing operator is applied to the matte. 3D reconstruction is based on stereo matching of the face regions in two camera pairs with a narrow baseline. Images from each pair are rectified at every frame using the approach by Fusiello et al. [38]. The rectification aligns corresponding epipolar lines into the same rows in both images to simplify the correspondence search. A disparity map encoding the correspondences contains single-value horizontal disparities and is estimated only for the reference view in each camera pair (lower camera in the capture setup).

The correspondence search between the rectified reference image $\hat{I}_t^r$ and matching image $\hat{I}_t^m$ is formulated as an energy minimisation problem. Equation 7.1 defines an energy function E for a disparity map D in $\hat{I}_t^r$ which consists of a data term and smoothness term.

$$E(D) = \sum_{\mathbf{p}\in P} \overline{NCC}\left(G_r(\mathbf{p}), G_m(\mathbf{p} - D(\mathbf{p}))\right) + \sum_{(\mathbf{p},\mathbf{q})\in V} \lambda |D(\mathbf{p}) - D(\mathbf{q})| \tag{7.1}$$

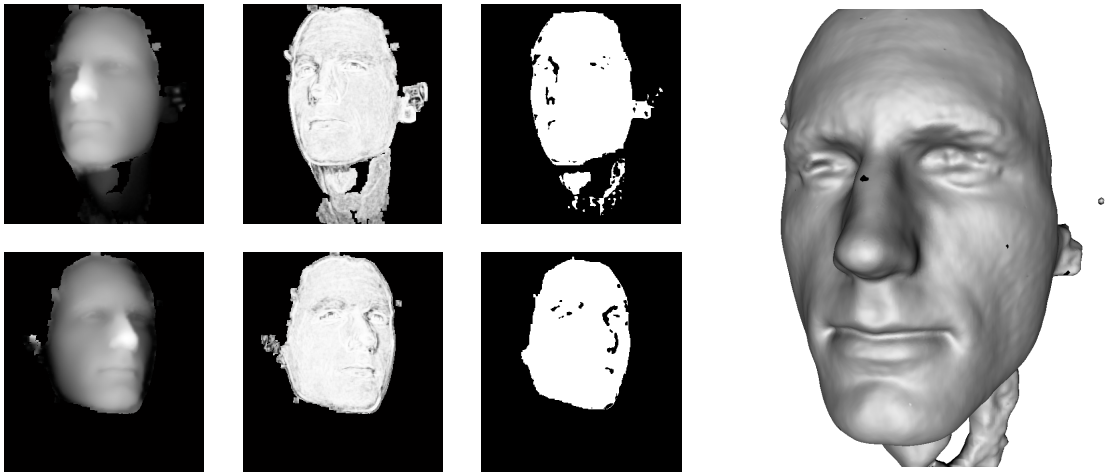
The map D contains a horizontal disparity $D(\mathbf{p}) = [d, 0]^T$ for each pixel $\mathbf{p}$ in a binary mask P which defines the face region in $\hat{I}_t^r$. The data term is based on a comparison of two sets of image points $G_r(\mathbf{p})$ in $\hat{I}_t^r$ and $G_m(\mathbf{p} - D(\mathbf{p}))$ in $\hat{I}_t^m$. The sets are square windows with a size s_G centered around the pixel $\mathbf{p}$ and its potential correspondence $\mathbf{p} - D(\mathbf{p})$. A matching cost $\overline{NCC}$ is an inverted normalised cross-correlation of grey-scale values: $\overline{NCC} = 1-(NCC+1)/2$. NCC has been chosen for its robustness against different camera gain between views unlike SSD. The smoothness term is expressed over the pairs of adjacent pixels $(\mathbf{p}, \mathbf{q})$ where a set V defines 4-point neighbourhood over

P. An absolute disparity difference between $\mathbf{p}$ and $\mathbf{q}$ is linearly penalised according to a smoothness coefficient λ. This enforces global smoothness of D without support for disparity discontinuities which is a reasonable assumption for the face.

The energy function in Equation 7.1 is minimised by a graph cut as in [90]. The linear smoothness term allows construction of a volumetric graph with 3D grid topology in the disparity space. A single minimum cut on this graph optimises E and yields a disparity map D for all pixels in P. The minimum cut is calculated by a maximum flow algorithm due to the duality of the problem. The augmenting-path algorithm by Boykov et al. [18] is used because it is optimised for graphs with a grid topology. Computational time and memory consumption for the graph cut calculation increase with the size of the graph, therefore the matching is constrained in several ways. The pixel set P is sampled by the regular grid with a step of 4 pixels in $\hat{I}_t^r$ and the disparity is quantised to integer values. The graph is further constrained by the face regions in both images which define a visual hull envelope for a surface of the face. Lastly, the disparity range $\langle d_{min}, d_{max} \rangle$ is given by the nearest and the furthest surface point from the camera pair which are selected in both views by a user.

The resulting D with integer disparities suffers from noticeable quantisation in the depth. This is refined by an additional graph cut inside a thin layer around the original integer solution given by an offset ± 1 disparity. Within the layer the resolution of the disparity is increased to 8 sub-pixel levels and the smoothness coefficient λ is proportionally divided by 8. Because an estimate of D is available, a square window G_r can be matched to a image point set G_m with adaptive shape. For a pixel $\mathbf{p}$, 2D sample points of $G_r(\mathbf{p})$ are mapped into $\hat{I}_t^m$ using the estimate of D around $\mathbf{p}$. This creates a new shape of sample grid G_m which is used for the whole range of disparities tested at $\mathbf{p}$. The adaptive shape of G_m improves correlation with G_r in comparison to square windows which leads to more precise calculation of disparity. To avoid the quantisation of integer D bias a new sub-pixel solution, D is smoothed by Gaussian kernel before using for the adaptation of G_m. The new disparity map refined by the additional cut is smoother and contains more detail at expense of small computation overhead.

The amount of detail in D is influenced primarily by s_G and λ. Large windows over-smooth fronto-parallel areas of the surface and create steps in the slanted areas but small windows yield a noisy result. The coefficient λ also influences the smoothness of the surface but on more local scale than the size of correlation window. Overall, the technique provides fairly accurate disparity map for plain skin because there is enough skin texture visible in HD resolution. A face with uniform white make-up lit by R, G, B lights does not pose a problem because the illumination emphasises the fine skin structure leading to rich texture information. The quality of matching is improved under these conditions comparing to the plain skin under white illumination and the facial shape is smoother with clearer details.



(a) (b) (c) (d)

Figure 7.4: Disparity maps from both stereo pairs (a), correlation score maps (b), disparity masks (c) and a resulting mesh created by the Poisson reconstruction (d). Brighter colour means larger disparity in (a) and higher correlations score in (b).

Each disparity map has a map of matching scores associated with its disparities (Figure 7.4(a,b)). This score map is thresholded to mask out areas with low accuracy such as parts observed at an acute angle (Figure 7.4(c)). The disparity map is filtered according to this mask and turned into a cloud of oriented points. The point clouds from both stereo pairs are merged into a single mesh M_t^g by the Poisson surface reconstruction [60] (Figure 7.4(d)). The quality of M_t^g is improved with pre-filtering of disparity maps. The reconstructed facial shape contains skin folds and larger wrinkles. Because the 3D reconstruction is performed independently at each frame, there is subtle surface bump-

iness changing over time. Also, spike artefacts occasionally appear due to instability of the Poisson reconstruction in some regions. However, this does not cause problems for the subsequent surface tracking because $\{M_t^g\}_{t=1}^T$ is used just as a shape prior.

7.4 Non-sequential surface tracking

Dense tracking of a facial surface over time is based on multi-view image sequences $\{\{I_t^c\}_{c=1}^C\}_{t=1}^T$ and unaligned mesh sequence $\{M_t^g\}_{t=1}^T$. A dissimilarity derived from a sparse set of points tracked throughout the sequence is used to calculate the traversal through the input data (Section 6.4.1). A typical set of points includes around 14 facial features which motion represents well changes of facial expression (Figure 7.5). These points are selected by a user and small amount of manual landmarking is necessary to train linear predictor tracker for them. Once the dissimilarity matrix is computed from the 3D point trajectories, a cluster tree is obtained using the algorithm in Section 6.3.3. Afterwards, it is expanded across cuts to allow multi-path temporal fusion (Section 6.5).

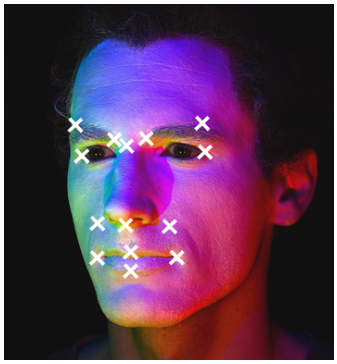


Figure 7.5: A sparse set of points tracked for dissimilarity computation.

Given the traversal tree, a user needs to define a topology of the mesh M_r which is tracked from the root frame throughout the input sequence. To simplify the task, the user designs a coarse mesh M_r' ($\sim$ 180 vertices) in 2D using the images from the root frame r. The vertices are placed onto the face in one view for each side and topology is defined by creating 2D mesh among their positions. The 2D positions are back-projected to the 3D space using depth maps of the raw mesh M_r^g rendered in each

view. The mesh M'_r is then iteratively subdivided to reach the resolution required. In each iteration mesh faces are uniformly subdivided into four triangles and new vertices are conformed to M^g_r. Position of new vertices are bilinearly sampled from the depth map of the view which the vertex normal faces the most. The resulting mesh M_r is tracked along branches of the traversal tree using robust image-oriented alignment from Chapter 5. The temporally consistent mesh sequence $\{M_t\}_{t=1}^T$ is obtained by the fusion of multiple tracking hypotheses around cuts.

In contrast to the results presented in Chapter 6, the surface tracking operates under colour illumination and an actor's face is covered by uniform white make-up. Intuitively, the uniform make-up should complicate the tracking process, such that visual markers are required to aid. However, colour illumination forms a strong texture from fine skin detail which has a similar amount of variation as the texture of plain skin lit by white lights. The actual complication is time-varying appearance of the face which is more severe under the colour illumination rather than white one. Also, shadows cast by directional lights move on the face over time. These temporal changes are handled by the adaptive texture of surface patches used in the frame-to-frame alignment. Moreover, the matching cost between images and the patch texture in Equation 5.1 uses colour information instead of grey-scale. The NCC is calculated separately for each channel and averaged afterwards. Longer computation is outweighed by better motion estimates. Previous approaches [68, 119] used normal maps computed by photometric stereo for the alignment. But experiments have shown that our technique achieves better results using the original images with colour illumination. The quality of surface tracking is similar under these conditions to the plain skin lit by white lights.

7.5 Geometric detail capture

Fine skin geometry which is not present in the aligned mesh sequence $\{M_t\}_{t=1}^T$ is captured by PSCL (Section 3.2). High-detail normal maps of the face are computed at every frame for one view in each stereo pair. To improve the quality of normals, the make-up is applied on the face as explained in Section 3.5.1. Because the capture system is focused on accurate reconstruction of the facial shape, the loss of actual ap-

pearance of the actor is acceptable. Although the sequences of normal maps contain high-quality skin detail, they suffer from several kinds of artefacts. Firstly, shadow regions contain normals with incorrect orientation because of missing constraints from one or more lights which are occluded by the other parts of the face [49]. Secondly, there is a weak low-frequency bias in the whole normal map due to errors in the photometric calibration [110]. Thirdly, normals are noisy in dark regions which have low SNR. These imperfections are corrected using the available mesh sequence $\{M_t\}_{t=1}^{T}$.

7.5.1 Normal map correction

Shadow correction in previous work often requires four or more lights and is tailored for standard PSWL [8, 24]. The technique for PSCL by Hernandez et al. [49] optimises the whole normal map such that integrability of the gradient field and smoothness of pixel colours are enforced in shadow regions. A simpler local approach by Brostow et al. [20] assumes a constant albedo which reduces uncertainty of the normal to two possibilities if one colour light is occluded. The final orientation of the normal is selected according to neighbouring unshadowed area. The proposed method exploits availability of medium-scale facial shape M_t which allows per-pixel normal correction assuming spatially varying grey-scale albedo. This is more suitable than the constant albedo assumption because brightness of the make-up varies due to uneven application. The base mesh provides a solution for multiple occluded lights as well.

Segmentation of shadow regions relies on the fact that each colour illumination is almost exclusively captured by the respective sensor. Thus, shadow masks for red, green, blue lights are obtained separately from each channel of an input image (Figure 7.6(b)). The segmentation uses similar rule as in [48] but it is performed on per-pixel basis instead of global Markov random field (MRF) optimisation. A pixel is classified as shadowed from the light (similarly for green, blue light) if a ratio between colour component and full vector $c_r/|\mathbf{c}|$ is under 0.15. To correctly include multiple-shadow regions and regions with originally dark appearance, the red light is also considered occluded for the pixel if $c_r < 17$. A raw shadow mask sequence from the pixel classification is noisy over time. Therefore, morphological opening operator with the size $5 \times 5 \times 5$ pixels is

applied across spatio-temporal volume to clean the segmentation. Finally, the shadow regions are dilated by 3 pixels to include their soft boundaries. An example shadow map for all three lights is depicted in Figure 7.6(c).

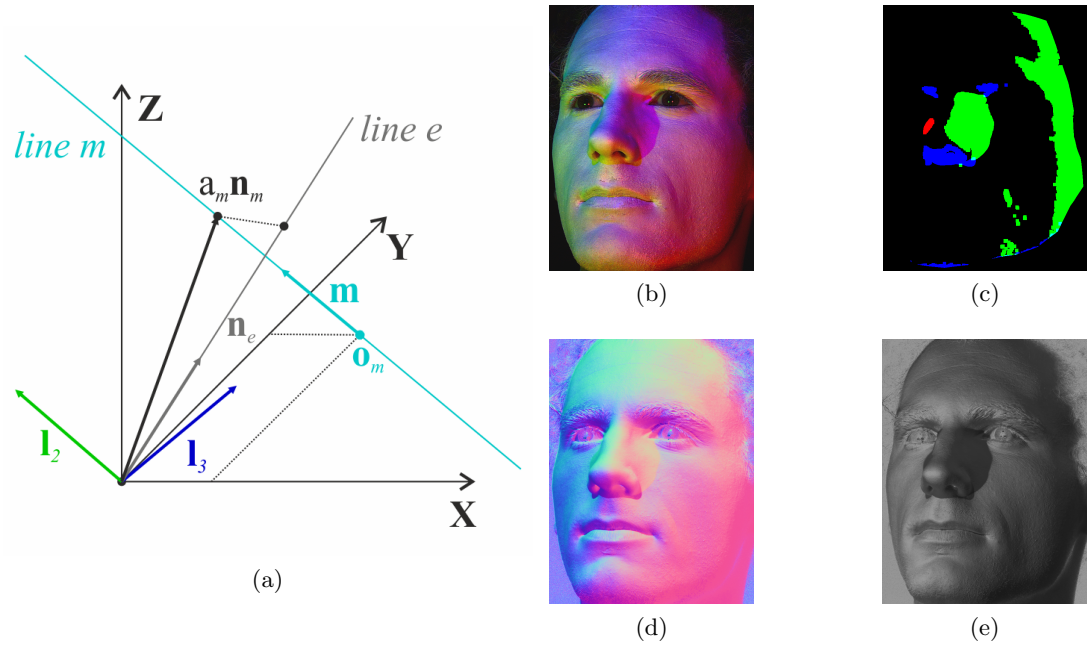


Figure 7.6: A corrected scaled normal $a_m\mathbf{n}_m$ calculated from a base mesh normal $\mathbf{n}_e$ and constraints by green and blue light (a). A line m is derived from a pixel colour, green light direction $\mathbf{l}_2$ and blue light direction $\mathbf{l}_3$. An input image (b) with its shadow map (c) where a colour indicates the light occluded. An original normal map (d) computed using PSCL from (b) and its rendered version lit by one frontal light (e). The shadow-like regions are the result of wrong normal orientation.

Normals in the regions with occluded lights have incorrect orientation because Equation 3.7 assumes contributions from all lights to an observed pixel colour. The normals are pushed away from the occluded lights which manifests as non-existent shadows (Figure 7.6(d,e)). Assume a single-shadow case where the red light is occluded for a particular pixel. After removing red-light components $\mathbf{v}_1$ and $\mathbf{l}_1$ from Equation 3.7, the linear system provides only two linear constraints from green and blue light. These constraints define a line m in the space of albedo-scaled normals (Figure 7.6(a)). A corrected albedo-scaled normal $a_m\mathbf{n}_m$ lies on the line m as defined in Equation 7.2.

$$a_m\mathbf{n}_m = \mathbf{o}_m + v\mathbf{m} \qquad v = \frac{((\mathbf{n}_e^T \cdot \mathbf{m})\mathbf{n}_e - \mathbf{m})\mathbf{o}_m^T}{1 - (\mathbf{n}_e^T \cdot \mathbf{m})^2} \tag{7.2}$$

The direction of the line m is denoted $\mathbf{m}$ and the point $\mathbf{o}_m$ is an intersection between m and xy-plane. A normal $\mathbf{n}_e$ from the base mesh M_t can be used as an additional constraint (Figure 7.8(a)). This is represented by a line e with the direction $\mathbf{n}_e$ starting from the origin of WCS in Figure 7.6(a). The coefficient v in Equation 7.2 defines a point on the line m which is the closest to the line e. This is taken as an end point of the corrected $a_m\mathbf{n}_m$. Note that only the unit normal $\mathbf{n}_m$ is stored in the corrected normal map. Correction of normals using M_t is applied independently to all pixels in the single-shadow regions. This results in three additional normal layers correcting for each light as visible in Figure 7.7(c-e). One extra layer is added for regions with multiple lights occluded or regions with very dark appearance where the incorrect normals are replaced by the mesh normals.

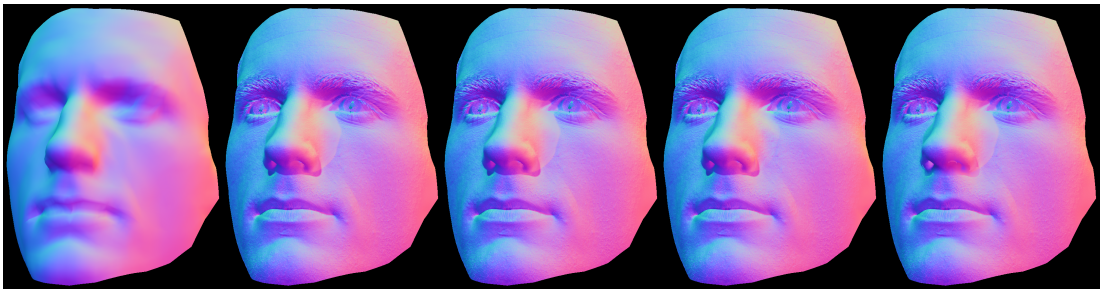


(a) (b) (c) (d) (e)

Figure 7.7: A shadow map (a) for the original normal map (b). Shadow-corrected normal layers for red, green and blue light (c, d, e) are overlaid onto (b) according to the segmentation in (a). This shows visible seams between the corrected regions and the original normal map due to different low-frequency bias.

The original normal map and the new shadow-corrected layers contain weak low-frequency bias. This is eliminated by the first stage of Nehab's technique [77] which conforms overall orientation of the normals to the mesh M_t. A photometric normal map and the mesh normal map (Figure 7.8(a)) are equally smoothed by Gaussian kernel (81×81 pixels, $\sigma = 13$ pixels) to acquire a low-frequency component from both sources. Subtraction of the photometric normals from their smoothed version yields high-frequency geometrical information such as skin structure and wrinkles. These details are transferred via a rotation field onto the smoothed mesh normals to form a high-detail normal map without the bias. The bias correction is applied separately to the

original normal map (Figure 7.8(c)) and each shadow-corrected layer (Figure 7.8(c-e)) because they are biased differently. The multiple-shadow layer does not need the correction because it contains mesh normals. This procedure makes all normal layers consistent with the mesh and each other and hence improves their subsequent fusion.



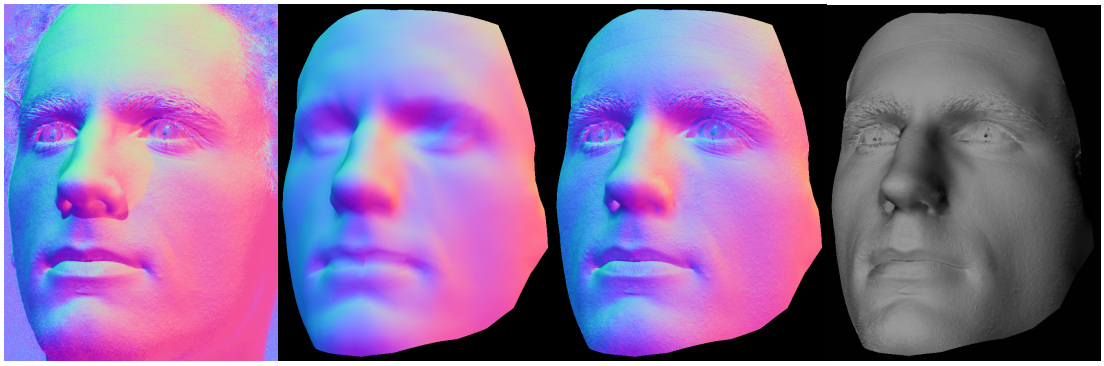
(a) (b) (c) (d) (e)

Figure 7.8: A normal map of the base mesh M_t (a) used to eliminate bias from the the original normal map (b) and shadow-corrected layers for red, green and blue light (c, d, e). The additional layers are overlaid onto (b) to show better consistency of shadow-corrected regions with the rest of normal map after the bias correction. Also notice different overall orientation of normals compared to Figure 7.7.

Finally, all normal layers are fused together according to the shadow mask into the final normal map (Figure 7.6(c)). Individual regions are linearly blended with each other over a 10-pixel range to prevent visual seams. The blending range requires the results of corrections bit beyond the region boundaries in the shadow mask. The resulting normal map sequence is constrained to the face area covered by the aligned mesh sequence. Figure 7.9 shows bias-free alignment of the corrected normal map with the base mesh and elimination of shadow artefacts.

7.5.2 Normal mapping

The corrected normal maps are back-projected from their views onto the mesh sequence $\{M_t\}_{t=1}^{T}$ at every frame (similarly to Section 5.5.3). Both maps, one for each side of the face, are merged together in a common UV domain of the mesh. This domain is created by unwrapping the reference mesh M_r onto 2D plane using a least squares conformal map in Blender [16]. Because the mesh topology does not change over time, texture coordinates of the vertices are fixed and only content of the UV space varies



(a) (b) (c) (d)

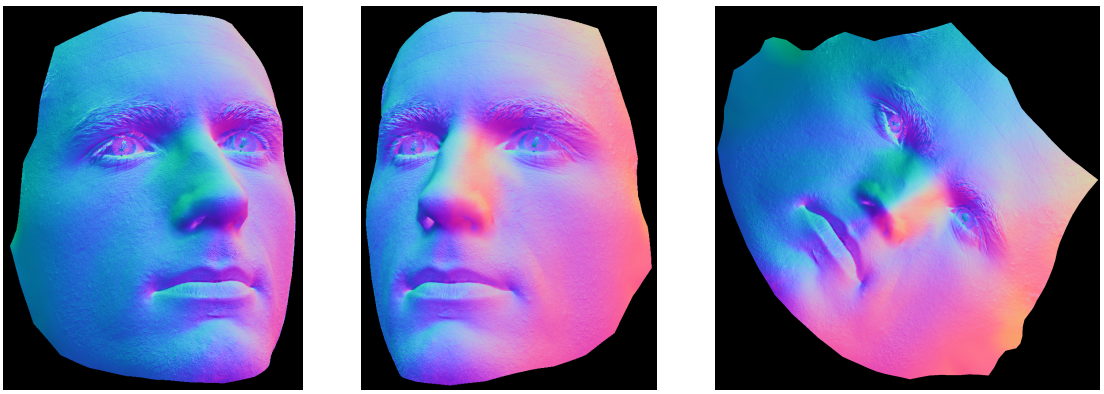
Figure 7.9: An original normal map (a), a mesh normal map (b), the resulting corrected normal map (c) and its rendered version lit by one frontal light. The rendering shows well-lit face without phantom shadows as expected from the frontal illumination in contrast to Figure 7.6(e).

between frames. At every frame the normals for each half of the face in the texture space are sampled from normal maps in respective side views (Figure 7.10). Although the normals are reasonably consistent across views because of bias correction, linear blending is applied across the border between halves of the face in the texture space (7-pixel range) to ensure completely seam-less transition. The resulting sequence of UV normal maps can have an arbitrary resolution. But to preserve the amount of skin detail observed in the image sequences, the effective area of the normal map should have similar resolution to the facial region in the images.

Facial performance is captured as a temporally consistent mesh sequence textured by high-resolution normal maps. The meshes provide medium-scale facial geometry and the normal maps fine skin geometry. The geometric detail is temporally aligned up to vertices of the base mesh and within mesh faces the alignment is bi-linearly interpolated. Precision of the alignment can be improved by increasing resolution of the mesh at expense of computation time. However, the interpolation does not cause any visible errors at the medium resolution used. The temporal consistency enables editing of the geometry over time. Any modification of the mesh at any frame can be propagated across the sequence exploiting vertex correspondence. Similarly, any modification of normals can be propagated between frames exploiting the fixed texture area in the common UV space. Rendering of the 3D facial model is primarily based on the normal

map which influences shading of the surface but the mesh provides underlying shape.

Alternative representation of the facial model is to include all geometrical detail into the mesh sequence at the expense of high mesh resolution [19, 13]. This leads to a much higher storage footprint than the combination of a medium-resolution mesh and a high-resolution normal map. The representation selected is also commonly used in the VFX industry because it is easier to work with. A typical facial animation rig deforms a mesh with a moderate number of vertices which drives various detail layers stored in a high-resolution 2D domain. Also, editing of geometric detail is simpler in the 2D UV space than the actual 3D space.

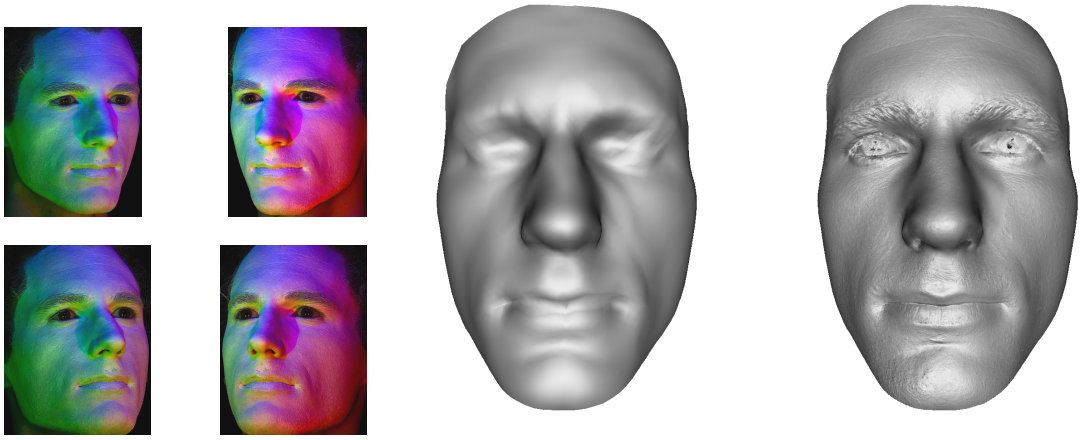


(a) (b) (c)

Figure 7.10: Corrected normal maps from side views for each half of the face (a,b) merged into a single UV normal map.

7.6 Evaluation

The system has been evaluated on performances of two actors (one male, one female). New datasets Martin-makeup1 and Alaleh-makeup1 captured with the colour lights and the make-up are described in Appendix G. Both are around 325 frames long ($13s$) and contain a variety of exaggerated expressions changing at a fast pace with small head motion. The resulting 4D performance models consist of a temporally consistent mesh sequence (2689 vertices and 5248 faces) and associated UV normal map sequence (1500×1500 pixels). Information about computation time through the processing pipeline is available for the dataset Martin-makeup1 in Appendix F.



(a) (b) (c)

Figure 7.11: Input images from 4 views (a), a temporally consistent mesh M_t (b) and the mesh M_t with a UV normal map (c).

Dissimilarity computation has required landmarking of 14 facial features (Figure 7.5) in 13 frames for linear predictor tracking. Stereo matching uses windows with a size $s_G = 15$ pixels and a smoothness coefficient $\lambda = 0.01$. The resolution of raw unaligned meshes is approximately 89000 vertices for Martin-makeup1 and 61000 vertices for Alaleh-makeup1. Robust frame-to-frame alignment has a common parameter configuration: NCC on colour, $d_o = 0.2mm$, $N_o = 11$, $w_g = 1.0$, $\sigma_g = 5mm$, $q_{lim} = 10mm$, $\xi_e = 0.15$, $\delta_e = 0.05$. The only difference between the datasets is in the smoothness coefficient s which is 0.5 for Martin-makeup1 and 1.0 for Alaleh-makeup1. Weaker regularisation for the male actor is due to the presence of more wrinkles. The cluster trees calculated for the non-sequential tracking are described in Table 7.1. The dataset Alaleh-makeup1 has more branched tree because the performance is less exaggerated and dynamic as in Martin-makeup1. Thus, there are more points with a similar expressions which encourage non-sequential jumps. This results in more cuts and a larger number of expansion nodes added for multi-path temporal fusion.

The final temporally consistent 3D models for both datasets are showcased by example frames in Figures 7.12, 7.13. Due to the dynamic nature of the results the reader is encouraged to watch the supplementary videos. The 3D models are rendered using a combination of OpenGL [61] and Cg shading language [80]. Projection of the base mesh onto an image plane is handled by OpenGL and final shading based on the normal

Dataset	β	Num. clust.	Num. br.	Min. br. l.	Avg. br. l.	Max. br. l.	Num. cuts	Num. ex. n.
Martin-makeup1	0.998	20	16	2	27.88	98	9	54
Alaleh-makeup1	0.99	37	28	8	19.36	39	15	90
DisneyFace	0.98	47	22	7	31.95	71	14	84

Table 7.1: Information about cluster trees for the datasets Martin-makeup1, Alaleh-makeup1 and DisneyFace: the granularity parameter β, number of clusters, number of branches, minimal branch length, average branch length, maximal branch length, number of cuts and number of expansion nodes.

map is produced by a shader programme using the Phong reflection model. The face is visualised with a uniform colour to demonstrate geometric detail and with a checker-board pattern to demonstrate temporal consistency. It is lit by a single directional light from the front and self-shadows are not modelled.

Temporally consistent mesh sequences presented in the video show correct base shape of faces without normal maps. The shape is recovered up to skin folds and larger wrinkles. The eyeballs and inside of the mouth are not properly modelled because of a view-dependent appearance which complicates the 3D reconstruction. The mouth interior is not included in the defined mesh topology at all, hence the smooth patch between the lips. It can be noticed in the video that the unaligned mesh sequences contains more detail than the aligned sequences such as better pronounced wrinkle shape. The reasons are the mesh resolution which is about an order of magnitude higher and filtering out some details together with outliers during motion regularisation.

Despite challenging performances the temporal alignment of meshes has high accuracy. The surface tracking is able to handle extensive deformations changing at a fast pace. This is demonstrated by a fixed checker-board texture locked down onto the face throughout the performance (Figures 7.12, 7.13(third row)). Another way of demonstrating the quality of temporal consistency is stability of normal maps in UV texture space over time. Noticeable swimming of the mesh occurs on the inner lips and around the eyeballs because they are undergoing the most complex motions. Also, there are parts of the face appearing and disappearing such as teeth and eyelids which are not explicitly considered in the surface model used for the tracking. Thus, blinks, eyeball or teeth movement are not properly estimated. Small drift is visible in some skin regions

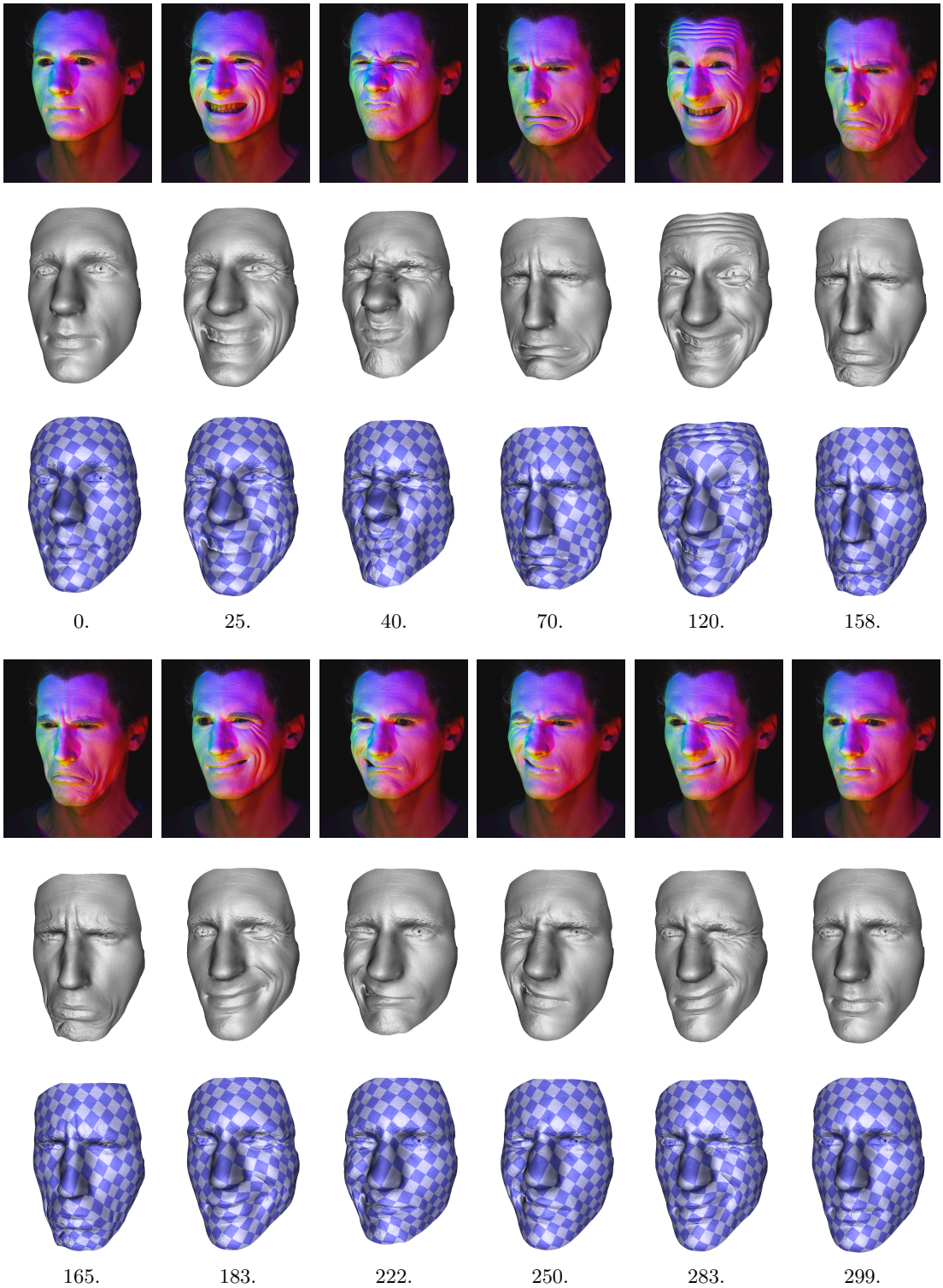


Figure 7.12: Snapshots from the temporally consistent 3D model for the dataset Martin-makeup1 - input images from one of the views (first row), meshes rendered with normal maps (second row) and meshes rendered with normal maps and a fixed UV texture (third row). The left most column represents the start/reference frame and the right most column the end frame of the sequence. Actual frame numbers are denoted.

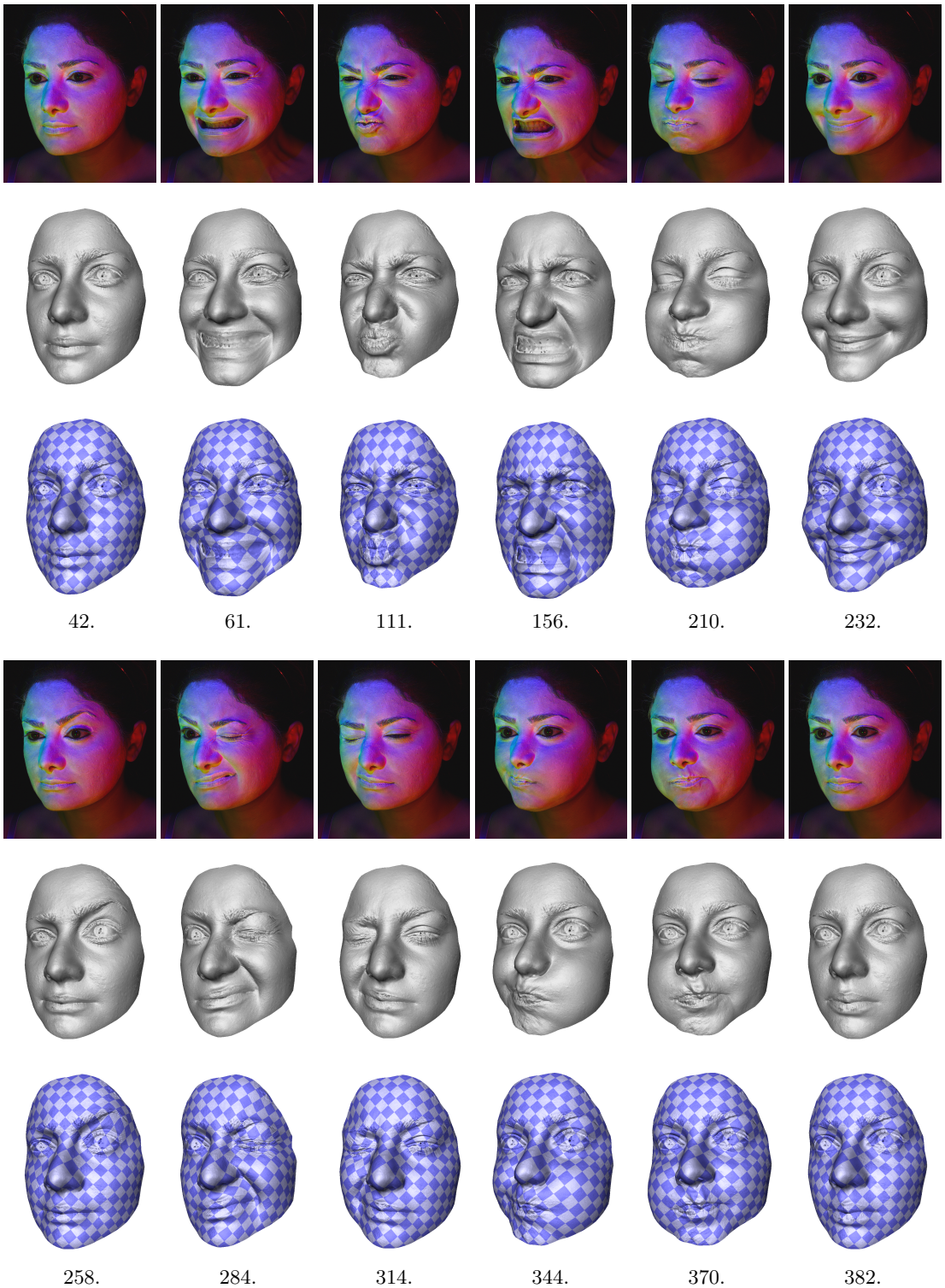
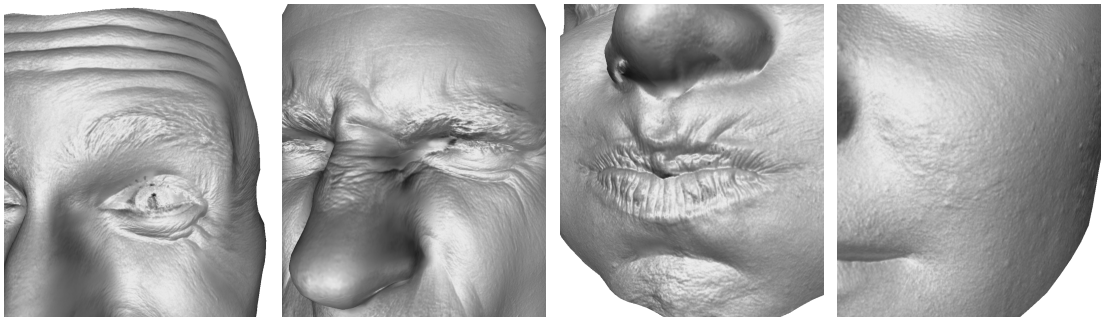


Figure 7.13: Snapshots from the temporally consistent 3D model for the dataset Alaleh-makeup1.

in a few cases when the fusion across cuts has to blend between significantly different alignment chains (noticeable during winks in Martin- makeup1).

Figures 7.12, 7.13(second row) illustrate correct reconstruction of fine details of the face in normal maps (magnified examples in Appendix E). Skin deformation and structure is captured to the full extent observed in original images. Close-ups in Figure 7.14(a,b) show rich skin wrinkling on the forehead and around the nose. Fine features such as eyebrow hair (Figure 7.14(b)) or creases on the lips (Figure 7.14(c)) are also present. The smallest skin details such as individual pores and blemishes are visible in Figure 7.14(d). There are also imperfections noticeable in the final models. Eyes (Figure 7.14(a)) and teeth (Figure 7.15(a,b)) have inaccurate and noisy normals because of their non-Lambertian properties. The geometric detail is well corrected in shadow regions but it appears smoother than surrounding area if light comes from a similar direction to the original occluded light source (Figure 7.15(c,d)). The smoothness coming from the underlying mesh is also visible when shadow segmentation occasionally includes parts of deeper skin folds (under the lower lip in Figure 7.15(a), on the lower cheek in Figure 7.15(b)).

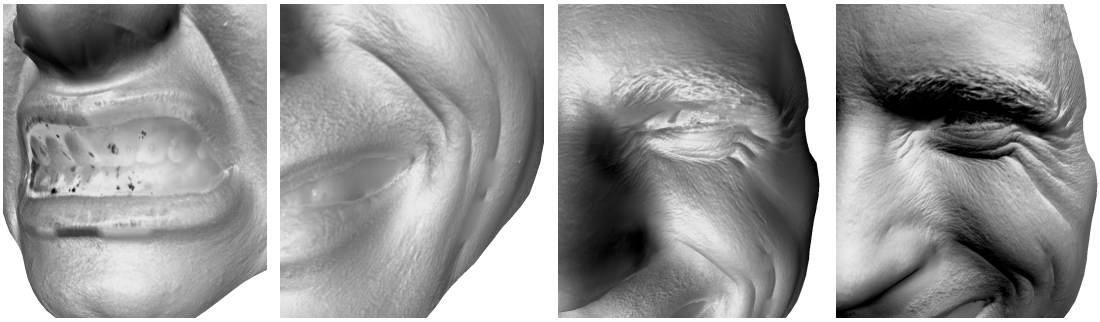


(a) (b) (c) (d)

Figure 7.14: Geometric details present in the final 3D models for the datasets Martin-makeup1 (a,b) and Alaleh-makeup1 (c,d).

7.6.1 Comparison to the state-of-the-art

The proposed system is compared to the state-of-the-art system by Beeler et al. [13]. Differences in key properties are listed in Table 7.2. Dissimilarity measure derived from



(a) (b) (c) (d)

Figure 7.15: Imperfections present in the final 3D models for the datasets Alaleh-makeup1 (a) and Martin-makeup1 (b,c,d) - wrong shape of teeth in (a,b); skin folds smoothed out under the lower lip in (a) and on the cheek in (b). The same region illuminated from different directions in (c,d) shows partial smoothness of corrected normals under former shadows. Areas on the side of face and around the nose appear smooth under one light direction and contain many details under the other direction.

3D trajectories of facial points reflects the actual motion of the face better than direct correlation among the images in [13]. Non-sequential traversal in [13] makes direct transitions from the root frame to all anchor frames and then sequential processing between the anchor frames. This can be seen as a suboptimal traversal tree in comparison to the cluster tree which considers a much larger set of possible transitions among frames. In the proposed approach, temporal alignment is interpolated for the skin details in normal maps from motion of the closest mesh vertices. This is not a source of noticeable inaccuracy in practice. Beeler et al. have direct alignment of all details because they are fully included in the temporally consistent high-resolution meshes. However, the solely mesh-based representation leads to large data size and it is more difficult to work with. PSCL provides metrically correct normals in contrast to an approximate skin structure derived from facial appearance by Beeler et al.

The results for the dataset DisneyFace have been released by Beeler et al. Because their system estimates the geometric detail using diffuse white illumination, it is not possible to make a comparison of final models with the proposed system which requires colour illumination. However, a partial comparison can be made for temporal consistency of the mesh sequence without quantitative assessment of geometric detail. This is performed in the same way as in Section 6.6.2, but using the cluster tree with multi-path

	Proposed system	Beeler et al. [13]
Capture conditions	directional colour illumination, make-up	**diffuse white illumination, no make-up**
Dissimilarity	**3D point set comparison**	global image correlation
Non-sequential traversal	**cluster tree**	anchor frames
Temporal consistency	interpolated alignment of details	**direct alignment of details**
Geometric detail	**metrically accurate**	approximate
3D model representation	**medium-res. mesh ($\sim$ 3000 vertices), high-res. normal map (1500×1500 pixels)**	high-res. mesh ($\sim$ 1200000 vertices)

Table 7.2: Comparison of the proposed system and the system by Beeler et al. [13]. Bold font marks an advantage over the other system.

temporal fusion. Also, resolution of the aligned meshes is higher (20000 vertices and 39810 faces) to show a result with finer facial shape. Dissimilarity is based on 15 facial features landmarked in 11 frames for linear predictor tracking. The high-resolution mesh sequence by Beeler et al. is used as a shape prior for surface tracking. The tracking has the following parameter configuration: NCC on grey-scale, $d_o = 0.2mm$, $N_o = 5$, $w_g = 10.0$, $\sigma_g = 5mm$, $U = 3$, $H = 2$, $q_{lim} = 5mm$, $s = 0.1$, $\xi_e = 0.15$, $\delta_e = 0.05$. In comparison to the experiments in Section 6.6.2 the parameters are adjusted to decrease the amount of computation per vertex and thus overall processing time for the denser mesh. Properties of the cluster tree used are described in Table 7.1.

A fixed texture overlaid on the aligned mesh sequence in Figure 7.16 demonstrates good-quality alignment. A few minor errors can be noticed such as small drift on the chin at the beginning or rougher inner lip contour on few occasions. The results are compared quantitatively to Beeler's result sub-sampled to the same mesh resolution. The overall difference computed as an Euclidean distance averaged across all vertices for all frames has mean = $0.24mm$, standard deviation = $0.313mm$. The spatial distribution of the difference is depicted in Figure 7.16 where lips and edges of the neck are the most dissimilar. Visually, both techniques achieve accurate temporal alignment

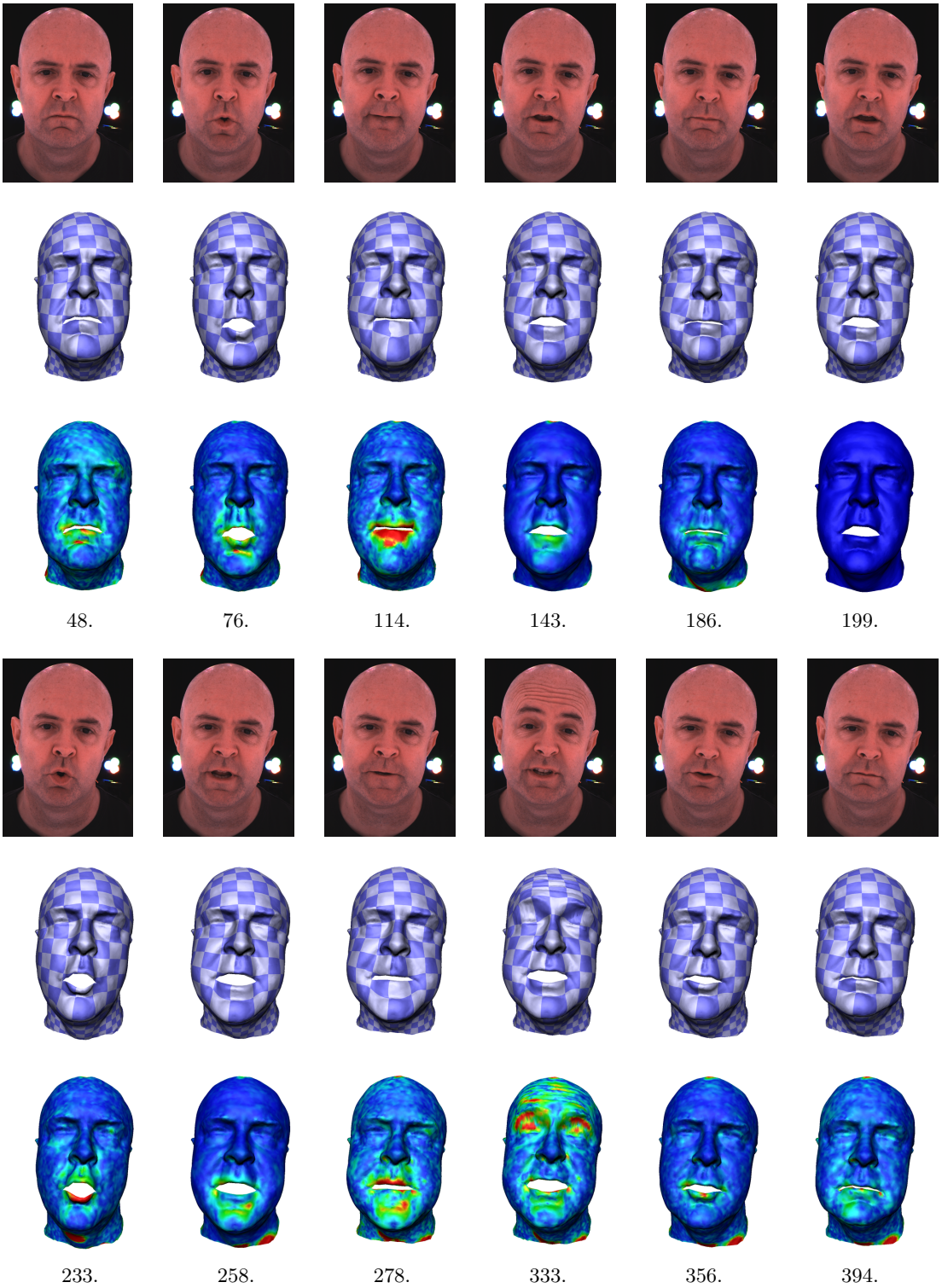


Figure 7.16: Snapshots from the temporally consistent mesh sequence for the dataset DisneyFace - input images from one of the views (first row), meshes rendered with a fixed UV texture (second row) and difference to the result by Beeler et al. [13] (blue = $0mm$, red = $2mm$) (third row).

of comparable quality (note that the discrepancy may be due to the errors in either approach). The proposed method suffers from slightly larger drift on the inner lips in some situations.

7.7 Conclusion

A novel 3D capture system for facial performance has been proposed in this chapter. This provides a full solution from a capture rig to rendering of final 3D models. A practical acquisition setup has been constructed from off-the-shelf equipment and does not require active illumination. Raw facial geometry used as a shape prior for surface tracking is reconstructed by a combination of stereo and Poisson reconstruction. A user-specified mesh is tracked non-sequentially by a image-oriented frame-to-frame alignment. New non-sequential traversal given by a cluster tree is introduced to the dense facial tracking.

The surface tracking framework is combined with PSCL which obtains normal maps on the face. It is demonstrated that the tracking works reliably under specific conditions required by PSCL (colour illumination, uniform make-up). The original normal maps computed directly from the images contain artefacts such as low-frequency bias, incorrect orientations in shadows and very dark areas. A correction process is proposed which exploits the available tracked meshes to improve the facial normals. The normal maps from several views are merged to a single time-varying map in the UV space of the aligned mesh sequence. This map contains metrically accurate skin structure which is coherent over time.

Output is a sequence of temporally consistent meshes and associated normal maps produced at standard camera rate. This representation of the performance naturally arises from the techniques used. However, it is also typically used in VFX industry because of smaller storage footprint and easier manipulation than a sequence of high-resolution meshes. It is possible to have a common representation across multiple separate performances of the same actor. Also note that there are no explicit assumptions about a face in the processing pipeline. Therefore, the system can be used for high-fidelity capture of other dynamic surfaces with large amount of detail.

Results demonstrate the ability of the system to acquire fine skin dynamics such as skin wrinkling, pore stretching etc. All details observed in the input image sequences is present in the final models. The system also achieves high temporal consistency of the facial models for challenging performances. Robust non-sequential tracking handles fast, complex changes of expressions in contrast to the previous sequential techniques. This approach also copes well with a substantial head motion and self-occlusions. The final model of a performance allows realistic rendering under different environmental conditions or space-time editing of the captured content. A comparison with the state-of-the-art system by Beeler et al. [13] shows similar quality of the results. However, the dataset used for the evaluation is not challenging enough to truly test capabilities of the both systems.

The main limitation of the approach is the make-up applied onto the actor's face which stems from the uniform chromaticity assumption of PSCL. The make-up is not necessary but greatly improves quality of the details recovered at the expense of the lost facial appearance. The correction of normal maps in shadow regions is too influenced by the mesh shape, such that the skin appears smooth under some light directions. The eyes and interior of the mouth do not have accurate stereo reconstruction and photometric normals due to their non-Lambertian reflectance. There is occasional minor drift of the mesh on the lips and in the eye sockets during complex deformations. Moreover, parts of the face not included in the mesh (e.g. the eyelids, teeth) appear and disappear which complicates the tracking. The temporal alignment of the geometric detail is given by the base mesh, hence it is interpolated inside mesh faces. However, this is not very noticeable in practice and can be alleviated by increasing the mesh resolution. Lastly, the skin normals are represented in WCS so they are not locally related to the base mesh. Because of that modification of the mesh does not automatically alter the normal map.

Chapter 8

Conclusions and future work

This thesis introduces a novel system for 3D facial performance capture including a full pipeline from capturing an actor to rendering a digital copy of the performance. The focus of this research is on achieving a high level of detail and temporal consistency for the geometric 4D model of a performance which is crucial for film production. This enables a change of viewpoint or relighting of the actor. The model of performance can be altered by space-time editing or can be used for building and driving a facial animation rig.

8.1 Conclusions

Chapter 3 demonstrates that photometric stereo with colour lights captures geometric detail up to skin-pore level. Time-varying normal maps are obtained at camera frame-rate and their quality is comparable to photometric stereo with white lights when uniform make-up is applied on the face. Photometric calibration of light-sensor-material interaction has been improved over Hernandez et at. [48]. Error analysis of the photometric stereo with colour lights formulates analytically the relationship between accuracy of the albedo-scaled normal and various input discrepancies (image noise, calibration error in light directions, calibration error in interaction between lights, sensors and a surface). This advances previous work in literature which has predominantly addressed image noise in standard photometric stereo with white lights. The analysis

also provides practical guidelines for constructing a capture setup, such that accuracy of the normal maps is maximised.

A baseline sequential surface tracking method using multi-view image sequences is presented in Chapter 4. This is based on the approach of Furukawa et al. [36, 37] where 3D matching of textured surface patches provides frame-to-frame 3D displacements for a template mesh. Motion of the mesh is regularised by weighted Laplacian deformation taking the displacements as soft constraints which gives more efficient linear solution than [36]. The objective function of 3D patch matching is empirically analysed for varying amounts of the surface texture to investigate limitations of previous techniques. This shows that the objective function is ambiguous for plain skin which results in drift for faces without markers or pattern. However, the baseline approach achieves accurate temporal alignment as [36] if a dense pattern is applied on the face.

A novel approach for robust sequential tracking is proposed in Chapter 5. This overcomes the limitations of the baseline approach for weak and time-varying texture which occurs in the case of plain skin without pattern make-up. 3D patch matching uses adaptive patch texture instead of track-to-first concept as [36]. The approach jointly optimises patches based on cooperative random sampling [6] which have not been applied to the surface tracking problem before. The estimated 3D displacements are constrained by a shape prior provided by per-frame stereo reconstruction. These extensions overcome the ambiguity of matching patches of plain skin over time. Clear advance on this type of data is demonstrated over the baseline tracking method (Chapter 4) and previous sequential 3D tracking methods [23, 79, 36, 37]. Sequential accumulation of frame-to-frame alignment errors over time still remains, especially during fast non-rigid movements of the face.

A non-sequential surface tracking framework is introduced in Chapter 6. This is one of the first non-sequential methods tackling drift in facial performance capture. Frame-to-frame alignments are performed along branches of a traversal tree which is defined over input frames. Different types of traversal trees can be calculated based on a dissimilarity measure between frames. A novel cluster tree approach is proposed which achieves improved tracking results over the minimum spanning tree [22] and shortest path tree

[52] previously used for whole-body tracking. The cluster tree takes into account the temporal order of frames allowing a balance between drift along tree branches and jumps where the branches meet. This greatly reduces drift and the impact of failure in comparison to sequential tracking [19, 119] and reduces the excessive number of jumps in comparison to other non-sequential traversals [52, 22]. Potential jumps are eliminated by multi-path temporal fusion between tree branches. The non-sequential approach also allows automatic global alignment of multiple performances by the same actor which has not been addressed by previous facial performance techniques.

The non-sequential framework is generalised to any frame-to-frame alignment method with an associated dissimilarity measure. Image-oriented alignment from Chapter 5 combined with dissimilarity based on sparse facial features achieves temporally consistent mesh sequences with very little drift. The temporal consistency of facial performances is comparable to non-sequential approach based on anchor frames [13]. Benefits of the non-sequential tracking are also demonstrated on cloth and whole body datasets. For whole-body performances, geometry-oriented alignment method with dissimilarity based on shape histograms [22] is used to show the flexibility of the framework.

Chapter 7 describes the whole 3D capture system for facial performance which combines methods from the previous chapters. A practical capture setup has been constructed using HD cameras with a standard frame-rate and static colour lights. Raw facial geometry used as a shape prior for surface tracking is reconstructed by a combination of stereo and Poisson reconstruction. Non-sequential surface tracking using the cluster tree (Chapter 6) is employed to obtain reliable temporal alignment for faces with uniform make-up under colour illumination which is required by photometric stereo (Chapter 3). Low-frequency bias and shadow artefacts in photometric normals are corrected by exploiting the underlying mesh. The normals from multiple views are merged to a single UV normal map sequence which textures the temporally aligned mesh sequence. Evaluation of the system demonstrates high-detail 4D geometric models of facial performances with accurate temporal consistency. The results are superior to other facial performance capture methods in terms of temporal alignment [19, 119, 39, 110] or accuracy of the facial geometry [19, 110]. Qualitative and quantitative comparison is made with the best state-of-the-art system by Beeler et al. [13] and temporal consist-

ency of 4D performance model has a comparable accuracy. The proposed system using photometric stereo has the advantage of metrically accurate surface normals.

This work represents a significant step towards practical 3D capture of facial performances. The state-of-the-art is advanced in high-detail capture of facial geometry and robust, accurate temporal alignment of facial performances. This is required for wider use of 4D performance models in film production.

8.2 Future work

High-resolution capture of geometric detail with the proposed approach requires the use of uniform make-up on the face. This is a consequence of the uniform chromaticity assumption for photometric stereo with colour lights. Hernandez et al. [46] propose a self-calibration method which optimises parameters for the dominant chromaticity of the surface. This could replace the current calibration which is less robust and practical. Vogiatzis et al. [110] extend previous work [20, 46] by replacing Lambertian reflectance with a Phong model but they still assume constant chromaticity. This improves results on faces without make-up but there is a bias in normals for non-dominant chromaticities. Anderson et al. [4] recently alleviated this problem by handling multiple piece-wise constant chromaticities. This has been evaluated for facial performance capture [3] but a limited number of chromaticities is not sufficient for high-quality normal maps. However, these techniques represent an interesting research direction which can completely eliminate the need for make-up and thus improve comfort of the actor.

Traversal of input data in non-sequential surface tracking is calculated from an approximate dissimilarity measure. The dissimilarity is only approximately proportional to the error of actual alignment between two frames. The relationship is non-linear which results in a sub-optimal traversal tree for the alignment method used. An interesting research avenue would be learning this relationship for a given combination of dissimilarity measure, alignment method and surface. The dissimilarity could be transformed by the learnt function and the calculated traversal tree and subsequent surface tracking would be improved.

Automatic calculation of the optimal traversal tree with respect to drift and jump tracking artefacts remains an open research problem. The cluster tree currently needs to be tuned for the granularity of frame clustering to achieve the best surface tracking. It would be beneficial to define a single measure for the traversal tree which explicitly describes the trade-off between potential drift and jumps based on dissimilarity. The optimal cluster tree could then be automatically selected omitting experimentation with surface tracking results. Furthermore, the measure could be used as an optimisation criterion for a new algorithm calculating a traversal tree in the dissimilarity space.

The presented surface tracking does not explicitly handle self-occlusions. If a region of the face becomes invisible to all cameras during a head motion, the related part of the template mesh starts to drift. This is because the patch matching error does not reliably indicate an occlusion and the patch texture incorrectly adapts to a different surface area. In the non-sequential framework, this failure is often limited to the period of occlusion if tracking approaches the occlusion from both directions in time. However, a more rigorous mechanism is required such as invalidation of patches based on sudden appearance change with respect to the past observations [91]. Another problem is re-detection of regions when they become visible again such that temporal consistency is preserved. Proper handling of self-occlusions would allow larger head motion with the same camera coverage and improve tracking of complex lip movements.

The problem of self-occlusions is related to the problem of appearing and disappearing parts of the face which are not included in the template mesh. The proposed approach does not model the eyelids or interior of the mouth and thus does not recover their motion. The user could create more detailed mesh including these regions but their tracking would require robust occlusion handling. Another research direction would be evolution of the 3D model of the face during the tracking as surface topology changes. Initial steps have been taken by geometry-based methods. Li et al. [64] do not alter the template mesh but evolve displacement maps over time as new details appear. A template-free method by Popa et al. [87] incrementally builds the surface mesh based on facial shapes occurring in a performance.

Capture of non-skin regions such as the eyes, facial hair, teeth is not directly tackled by the proposed system. Because of their complex properties shape reconstruction and motion estimation are not accurate. Specialised techniques targeting these features would substantially improve realism of the final performance model. An important step in this direction was taken by Beeler et al. [11] who introduced coupled 3D reconstruction of facial hair and skin for static capture of the face. This thesis also did not address facial appearance which is an important part of performance capture. However, advances in photometric stereo with colour lights could make it possible in the future. Alternatively, non-sequential surface tracking can be combined with other detail capture techniques such as shape-from-shading or Light Stage.

Facial performance capture has many research challenges left before creating a true digital double of an actor. Technologies developed in the process will have a great impact in film production and other areas related to understanding human faces.

Appendix A

Marker-based facial performance capture

This appendix presents an early version of facial performance capture system. The presented approach was the first to combine stereo 3D reconstruction with PSCL for facial performance application. Resulting 3D models have coarse temporal consistency based on motion of markers painted on the face.

A.1 Overview

The pipeline of the marker-based system is illustrated in Figure A.1. An actor with markers painted on the face is recorded using the capture setup described in Section 7.2. Input of the pipeline are multi-view image sequences of the performance.

Surface tracking: Coarse measurement of facial shape and deformation over time is extracted from the motion of point markers on the face. Linear predictor tracking provides 2D tracks of the markers in individual views. Triangulation of the tracks between views in stereo pairs is performed according to user-defined correspondence. Temporally aligned coarse meshes are constructed from 3D trajectories of markers given a mesh topology specified by the user.

3D reconstruction: Medium-level facial geometry is reconstructed at every frame using stereo matching in both camera pairs. The coarse mesh of the face is used to constrain the correspondence search and merge resulting disparity maps into a single surface. A dense model is created by iterative coarse-to-fine subdivision and refinement of the coarse mesh according to the disparity maps. Output is a sequence of dense meshes with coarse temporal alignment.

Geometric detail capture: Fine skin detail is obtained at every frame using PSCL with the aid of white uniform make-up on the actor's face. Normal maps from one view in each stereo pair are corrected using the dense mesh sequence. The corrected normals are mapped onto the dense meshes to produce highly detailed 3D models.

Output of the pipeline is a sequence of high-resolution 3D models capturing subtle dynamics of the performance such as skin wrinkling, pore stretching, etc. However, the temporal consistency of the models is approximate due to the sparse set of the markers.

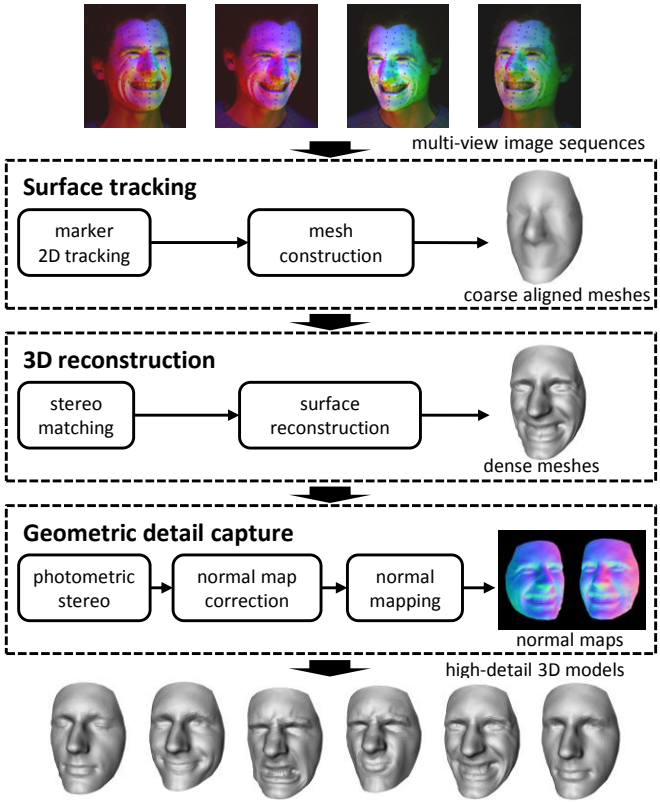


Figure A.1: A diagram of the processing pipeline for the marker-based capture system.

A.2 Surface tracking

A coarse temporally aligned model of facial performance is reconstructed from markers painted on the face. The markers are easier to track in comparison to arbitrary points on plain skin or uniform make-up. They are placed on the face such that their spatial positions provide a good approximation of the facial shape in any expression.

Initially, positions of the markers are manually selected in individual views at the first frame. Linear predictor technique [81] is used for sequential 2D tracking in each view. To handle drift and non-rigid deformations, this technique has a training stage requiring some manual landmarking (more explanation in Section 6.4.1).

Correspondence between the markers across views is established by the user during the landmarking. A 3D trajectory of a marker over time is reconstructed from corresponding 2D tracks in the closest camera pair. The trajectory can be noisy over time because of independent 2D tracking in each view. 3D positions of all markers at each frame are converted to triangle meshes according to the user-defined connectivity between the markers. The result is a sequence of coarse 3D meshes which are temporally aligned.

A.3 3D reconstruction

Disparity maps are computed for reference views in both stereo camera pairs as described in Section 7.3 with few differences. Firstly, segmentation of the face is automatically given by projection of the coarse mesh from markers into images. Secondly, the coarse mesh provides an estimate of disparity map which can warp sample grid in the matching image to improve correlation calculation. Thirdly, the coarse mesh is used to constrain a volume of interest in the disparity space for correspondence search. The 3D graph is constructed within a layer defined by an offset from the disparity map of the coarse mesh. The minimum graph cut computed on this reduced graph saves a considerable amount of time and memory compared to the reduction by a fixed disparity range.

Each stereo pair provides a sequence of disparity maps for its reference view. These two sequences are merged into dense meshes of the whole face at every frame using

the coarse mesh sequence. The coarse mesh is iteratively subdivided and conformed to both disparity maps as explained in Section 7.4. This process is repeated until the mesh captures well the facial shape present in the disparity maps. Figure A.2 illustrates reconstruction of the dense mesh in 3 iterations.

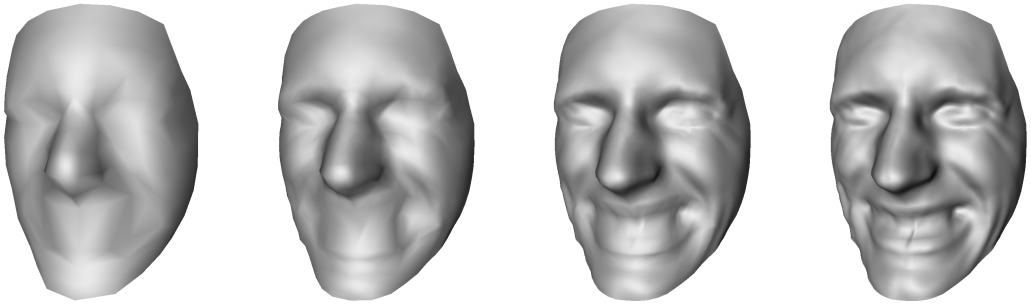


Figure A.2: Iterative reconstruction of a dense facial mesh using the coarse mesh among markers and disparity maps. The coarse mesh (left most) is subdivided 3 times to achieve the resulting dense mesh (right most).

The resulting sequence of dense meshes has the same resolution and topology in all frames due to the same number of subdivisions. But the temporal consistency of the sequence is approximate because only the initial coarse meshes are temporally aligned. The coarse temporal alignment is interpolated for new mesh vertices in the final meshes. Thus, there is no guarantee that corresponding vertices at different frames represent the same surface points.

A.4 Geometric detail capture

Geometric detail capture is similar to the approach described in Section 7.5. However, this is an earlier version of the method, so there are differences which led to latter improvement. Computation of initial normal maps is the same using PSCL aided with white uniform make-up. Artefacts in normal maps are corrected exploiting the dense facial meshes but the correction steps are bit different to Section 7.5.1.

Markers introduce many artefacts as their normals are very noisy due to dark appearance. They are segmented by special rule $|\mathbf{c}|/|\mathbf{c}_{white}| < 0.025$ within 12-pixel radius

from their 2D positions obtained during the tracking. Shadow maps are only morphologically dilated at each frame to include boundary pixels of individual regions. Thus, they are more noisy and less stable over time in comparison to Section 7.5.1.

Single-shadow regions corrected according to Equation 7.2 are linearly blended with their surroundings to prevent visual seams. The blending occurs within a 10-pixel range outside of the refined regions. Because there is different low-frequency bias in individual regions due to calibration errors, the blending transitions are noticeable in the resulting normal map.

Bias correction is performed once for the shadow-corrected normal map with blended regions instead of separate processing of normal layers from the shadow correction as in Section 7.5.1. The bias is eliminated by transferring high-frequency detail from the photometric normal map onto smooth normal map of the base mesh. The lowest frequency in the transferred detail is limited by the blending range in the shadow correction because the transitions between the regions should not be transferred. This causes a loss of medium-scale information in the bias-free normal map.

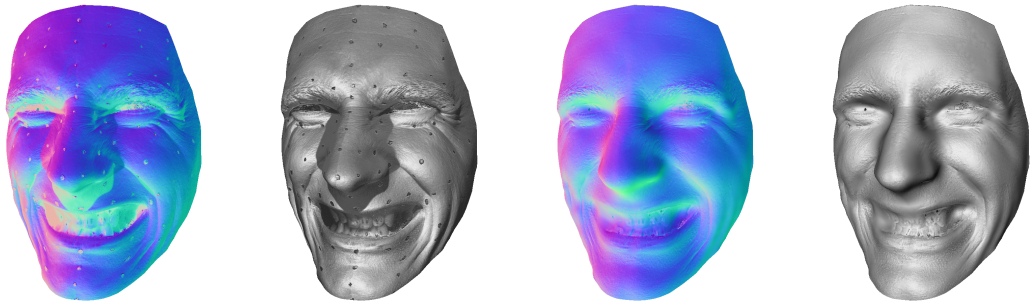
Lastly, incorrect normals in the multiple-shadow regions and markers are completely substituted by the normals from the underlying mesh. Because of the previous bias elimination, these regions can be consistently blended into the final normal map.

The detail capture produces two sequences of high-resolution normal maps for one view in each stereo pair. At every frame a dense mesh is projected to both views to obtain texture coordinates for the vertices. Each half of the face is textured by the normal map from the closer view. The 3D facial model consists of the dense mesh and a pair of the normal maps from camera viewpoints stored in one texture image (example would be Figures 7.10(a,b)). This is different from Section 7.5.2 where the normals are stored in a single UV normal map.

A.5 Evaluation

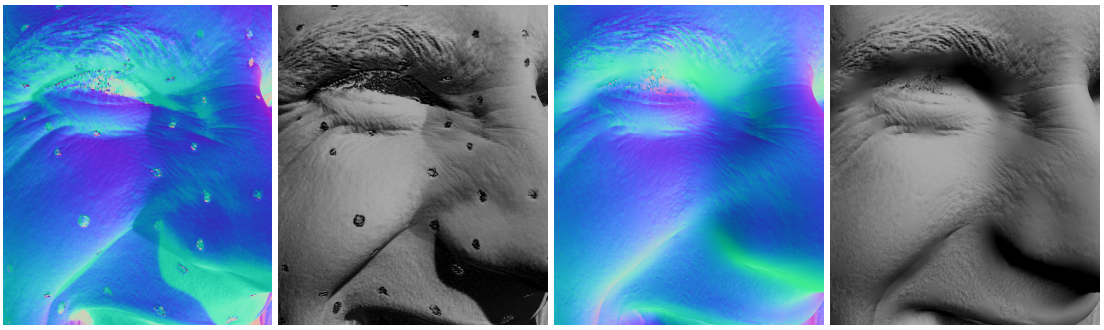
The dataset Martin-makeup2 (Appendix G) has been processed by the described marker-based approach. Figure A.6(first row) shows several frames from the captured perform-

ance in one of the views. The actor is painted with 142 markers and their 2D tracking requires manual landmarking in 9 distinct facial expressions in each view. Temporally aligned coarse meshes have 142 vertices and 254 triangles. Stereo matching is restricted to the volume within $2cm$ from the coarse meshes. Dense meshes are created by triple subdivision of the coarse sequence (8241 vertices and 16256 triangles).



(a) (b) (c) (d)

Figure A.3: A 3D model of the face at frame 214 from the dataset Martin-makeup2. The dense mesh is textured with the original normal maps by PSCL (a, b) and the corrected normal maps (c, d). Normals are colour-coded (a, c) and rendered under frontal directional light (b, d). One of the input images for frame 214 is in Figure A.6(first row).



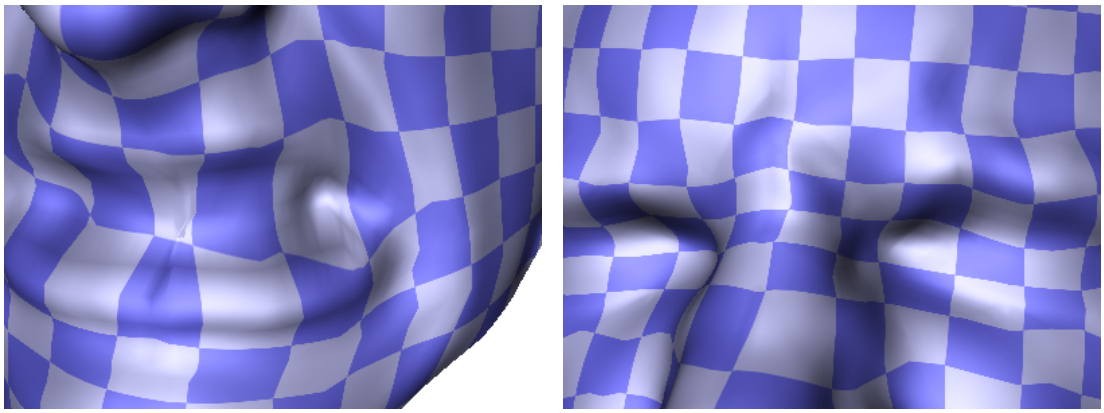
(a) (b) (c) (d)

Figure A.4: Close-up of the model in Figure A.3 to show the captured skin detail - original normal maps (a, b) and corrected normal maps (c, d).

Figures A.3 and A.4 demonstrate accurate facial shape up to skin structure at an example frame. Original normal maps by PSCL are compared to the outcome of normal correction which successfully eliminates bias in overall orientation, phantom shadows (e.g. around the nose) and markers. Figure A.4 shows the preserved skin detail in

shadow-corrected regions but partial smoothness propagated from the underlying mesh (e.g. under the nose). Smooth smudges at the place of markers are due to substitution with mesh normals.

Figure A.6 depicts 3D models of the face in different expressions from the captured performance. Larger skin details such as wrinkles look bit flat in comparison to the results of newer marker-less system from Chapter 7 (Figure 7.12). This is due to some loss of medium-scale shape information during normal map correction. Borders of the corrected regions are noticeable and not coherent over time. Deformation of a UV texture fixed to mesh topology in Figure A.6(third row) is correct for large facial movements. But the temporal alignment is approximate between positions of individual markers. This is noticeable as crooked sides of texture squares in Figure A.5.



(a) (b)

Figure A.5: Close-up of the meshes with a fixed UV texture at frames 102 (a) and 157 (b) in Figure A.6.

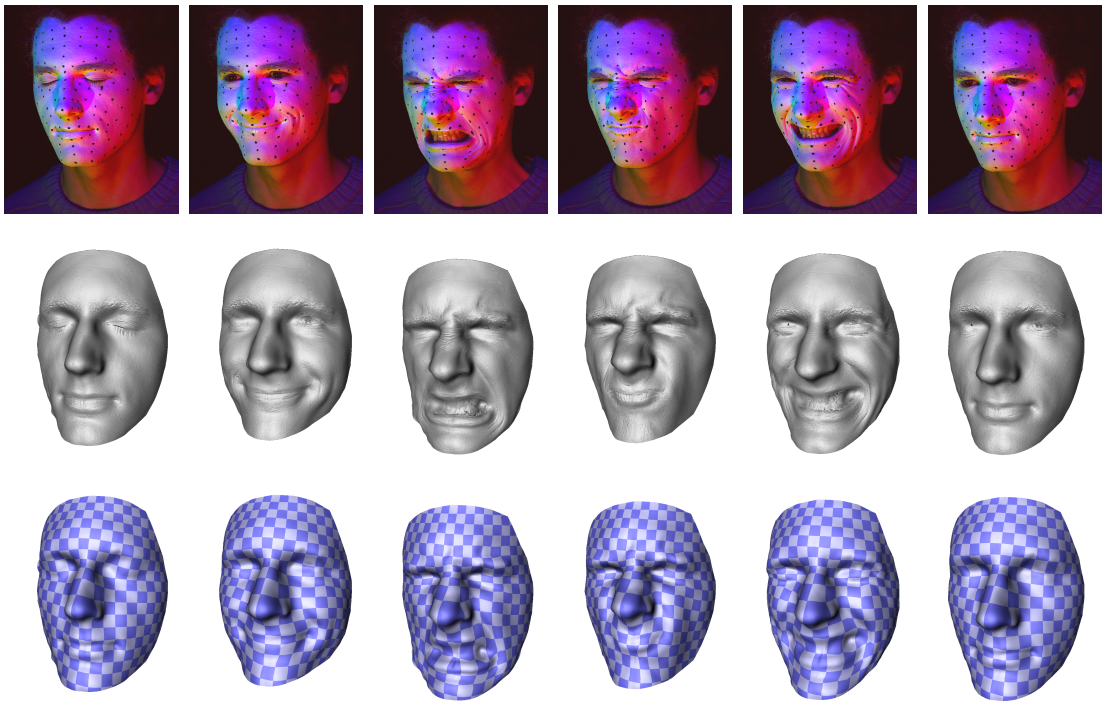


Figure A.6: Snapshots of 3D facial model for the dataset Martin-makeup2 - input images from one of the views (first row), meshes rendered with normal maps (second row) and meshes rendered with a fixed UV texture and without normals (third row).

Appendix B

Quantitative evaluation using a mirrored sequence

In Chapter 5 quantitative evaluation of temporally consistent mesh sequences is performed using SAD error on unwrapped surface textures (Section 5.5.3). Alternative quantitative measure proposed by Furukawa et al.[36] is presented in this section. This is used to compare the baseline and robust sequential tracking on the datasets Martin-pattern1, Martin-markers1, Martin-skin1 and Martin-skin2.

Evaluation of drift proposed in [36] does not use directly image information and focuses on the mesh sequence $\{M_t\}_{t=1}^{T}$. The input sequence of observations $\{O_t\}_{t=1}^{T}$ is reversed and concatenated to the original one (not repeating the last frame T). This creates a new sequence with $2T-1$ frames where the observations O_t are the same at the frames $T-\Delta t$ and $T+\Delta t$ (frame pair index Δt is from the range $\langle 0, T-1\rangle$). The mirrored sequence is tracked and the corresponding meshes $M_{T-\Delta t}$ and $M_{T+\Delta t}$ are compared across all frame pairs.

Comparison of the meshes uses an average Euclidean distance between corresponding vertices. Good temporal alignment is indicated by a small difference between the initial mesh M_1 and the last mesh M_{2T-1} after forward-backward pass through the performance. The errors for the frame pairs in-between demonstrate stability of the tracking over time. However, this measure cannot be seen as an estimate of tracking

accuracy for every frame because of the varying number of transitions between the frame pairs. The mesh distance generally enlarges with increasing pair index Δt because likelihood of accumulating errors increases with the number of transitions in between the frames. Only a difference between the meshes M_1 and M_{2T-1} can be considered as an accuracy for the frame T (the end of the original performance) since the same sequence of observations is tracked forwards and backwards as in the actual processing. Note that this approach cannot be used for non-sequential tracking in Chapter 6. The non-sequential tracking does not take the forward-backward path between the mirrored frames as assumed. Thus, the comparison of respective meshes does not reflect the amount of drift in between.

Figure B.1 shows graphs of average vertex distance with increasing Δt for the baseline and robust tracking method on mirrored datasets. The robust method performs worse overall on the dataset Martin-pattern1 because of small incremental drift (Figure B.1(a)). In comparison the distance is lower for the baseline method but it fluctuates more. This is caused by shakes of the mesh at extremes of the expressions. However, the maximal error for the robust approach after returning at the beginning of performance is below $0.9mm$ which demonstrates still good accuracy. For the more challenging dataset Martin-markers1 the robust technique yields approximately $1.5mm$ maximal vertex distance after gradual rise (Figure B.1(b)). The baseline technique has a similar increasing trend but the maxima are higher which shows higher instability of tracking. On the dataset Martin-skin1 the robust method yields slightly higher final error $\sim 1.7mm$ as for the markers (Figure B.1(c)). The baseline technique is clearly worse because severe mesh distortions appear early in the tracking of mirrored sequence. The robust method has a sharper increase of the vertex distance for the more complex dataset Martin-skin2 (Figure B.1(d)). This is caused by significant mesh distortions accumulated in the eye and mouth regions. The baseline method failed completely in the first half of the mirrored performance, thus it was not possible to make a comparison between the frame pairs.

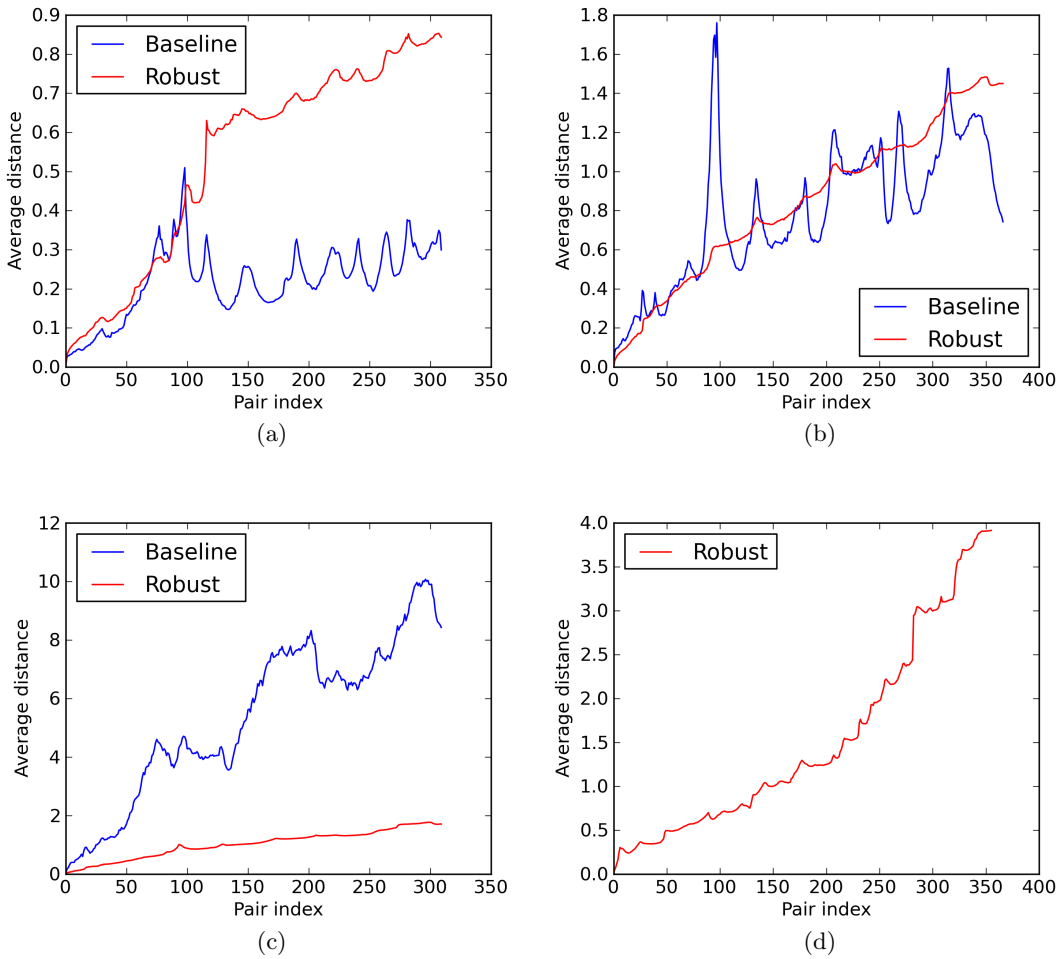


Figure B.1: Average vertex distance between corresponding frame pairs on the mirrored sequence for the datasets Martin-pattern1(a), Martin-markers1(b), Martin-skin1(c) and Martin-skin2(d). The unit is *mm*. Note that the baseline method failed for the dataset Martin-skin2.

Appendix C

Traversal trees

Non-sequential surface tracking is evaluated for different traversals through the input sequences of several datasets in Section 6.6. The following tables list the traversal trees sampled from the tree spectrum for each dataset. Each tree is described by several properties: β value, the number of clusters, the number of branches, average branch length and the number of cuts. Bold font style denotes the traversal tree resulting in the best temporally aligned mesh sequence.

β	No. of clusters	No. of branches	Average branch length	No. of cuts
1(SEQ)	1	1	354	0
0.9998	11	6	76.33	3
0.9992	21	16	45.06	8
0.996	31	27	18.3	19
0.99	**41**	**28**	**18.32**	**19**
0.97	61	27	21.07	17
0.94	83	30	20.7	20
0.9	97	32	17.97	24
0.8	135	37	19.11	32
0.7	175	38	19.87	34
0.6	213	38	19.18	34
0(MST)	355	41	19.24	38
SPT	-	354	1	352

Table C.1: Information about the evaluated traversal trees for the dataset Synthetic-skin1.

β	No. of clusters	No. of branches	Average branch length	No. of cuts
1(SEQ)	1	1	354	0
0.9998	11	10	63.9	5
0.9992	21	23	25	12
0.996	31	24	25.17	15
0.99	41	27	22.52	17
0.98	59	31	23.1	20
0.95	**79**	**31**	**22.71**	**19**
0.9	99	39	21.62	28
0.8	139	45	22.53	38
0.7	183	43	21.33	40
0.6	217	45	20.93	44
0(MST)	355	55	20.73	62
SPT	-	354	1	352

Table C.2: Information about the evaluated traversal trees for the dataset Martin-skin2.

β	No. of clusters	No. of branches	Average branch length	No. of cuts
1(SEQ)	1	1	345	0
0.9994	11	12	52.92	5
0.999	15	12	42.33	6
0.998	21	18	37.33	9
0.996	27	18	39	11
0.99	37	21	30.62	12
0.98	**47**	**22**	**31.95**	**14**
0.96	61	27	32.63	19
0.93	79	29	32.41	18
0.9	91	30	30.6	20
0.8	117	38	27.29	30
0.6	181	41	26.22	42
0(MST)	346	44	26.61	56
SPT	-	345	1	344

Table C.3: Information about the evaluated traversal trees for the dataset DisneyFace.

β	No. of clusters	No. of branches	Average branch length	No. of cuts
1(SEQ)	1	1	319	0
0.999	18	18	39.06	8
0.997	30	20	27.1	9
0.994	**40**	**27**	**31.44**	**16**
0.98	60	26	31.58	19
0.96	80	28	28.39	23
0.92	104	32	31.69	33
0.9	118	39	31.15	39
0.8	188	37	32.65	41
0(MST)	320	39	31.08	43
SPT	-	319	1	317

Table C.4: Information about the evaluated traversal trees for the dataset Garment.

β	No. of clusters	No. of branches	Average branch length	No. of cuts
1(SEQ)	1	1	1049	0
0.999	22	17	112.41	8
0.998	30	25	131.84	12
0.996	**42**	**29**	**127.03**	**16**
0.994	52	37	126.95	23
0.99	62	34	124.53	22
0.97	102	59	122.48	39
0.95	130	62	121.99	46
0.93	152	66	91.23	50
0.9	176	66	90	55
0.8	250	81	102.96	76
0.6	328	97	104.62	98
0(MST)	1050	138	74.69	249
SPT	-	1049	1	1047

Table C.5: Information about the evaluated traversal trees for the dataset StreetDance.

Appendix D

SEW, SPL and CUT measures

Section 6.6.7 in Chapter 6 defines SEW, SPL and CUT measures which describe different characteristics of traversal trees. The following graphs present experimental results for individual measures which are discussed in Section 6.6.7.

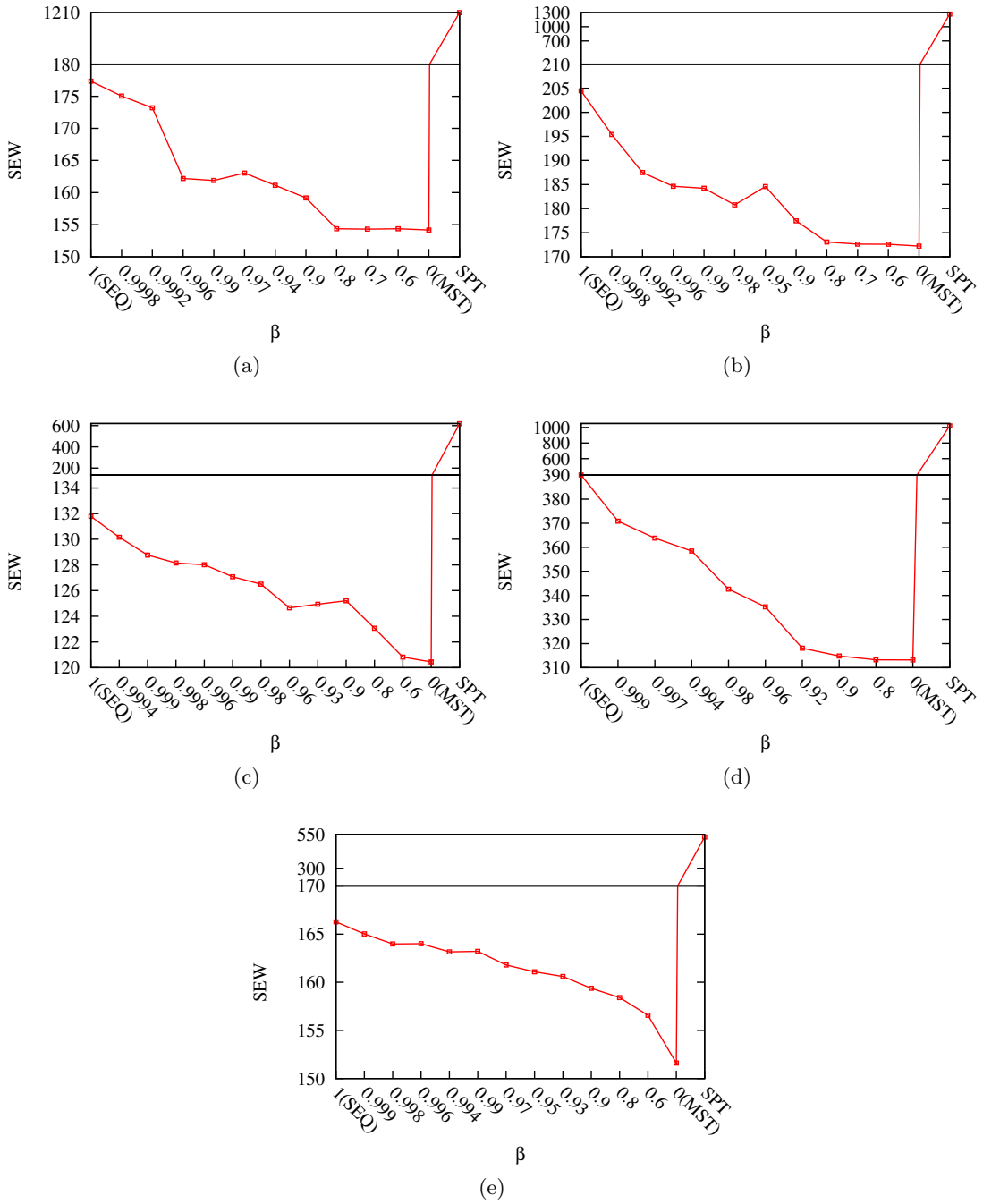


Figure D.1: *SEW* measure across different traversals for the datasets Synthetic-skin1(a), Martin-skin2(b), DisneyFace(c), Garment(d) and StreetDance(e). The best cluster trees empirically selected for Synthetic-skin1 ($\beta = 0.99$), Martin-skin2 ($\beta = 0.95$), DisneyFace ($\beta = 0.98$), Garment ($\beta = 0.994$) and StreetDance ($\beta = 0.996$).

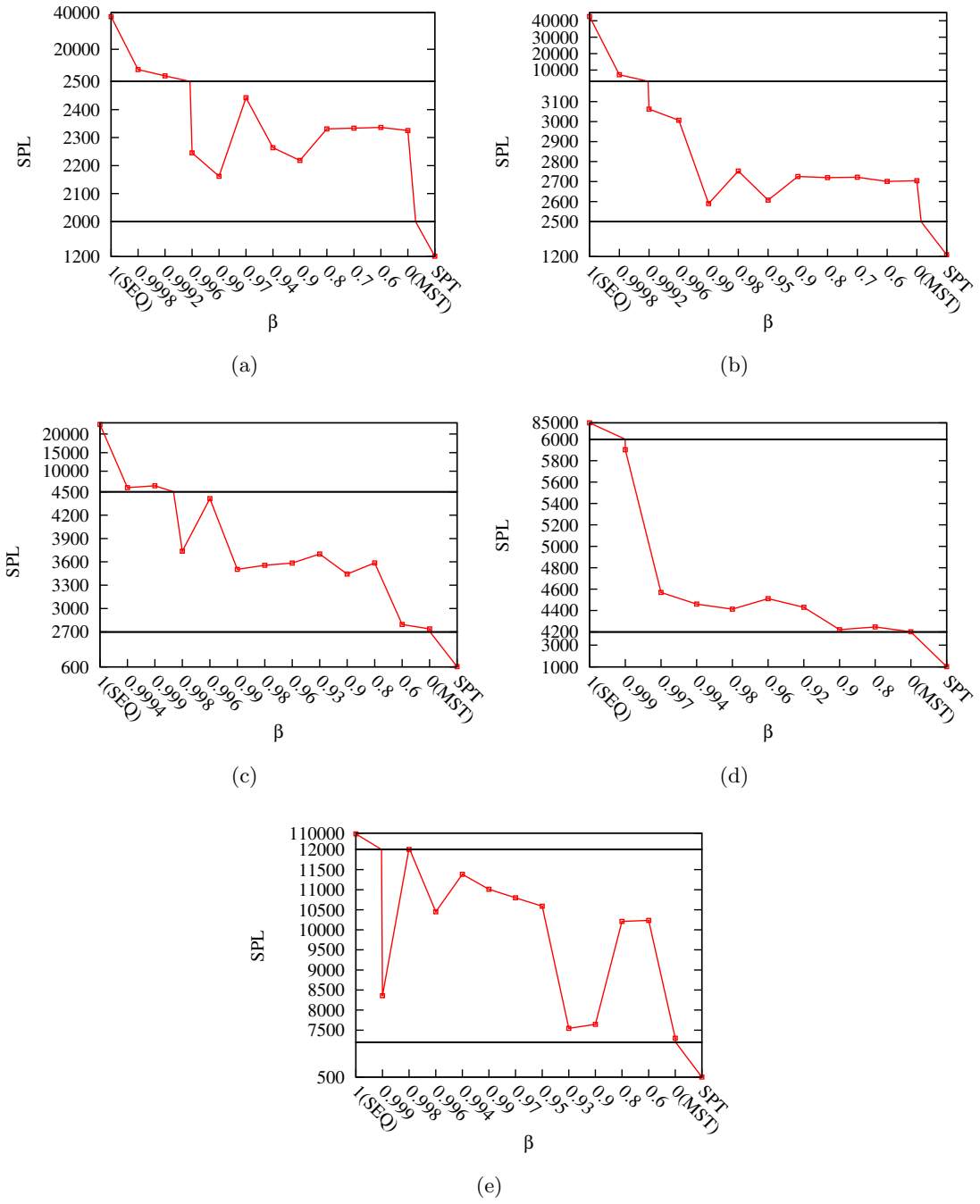


Figure D.2: SPL measure across different traversals for the datasets Synthetic-skin1(a), Martin-skin2(b), DisneyFace(c), Garment(d) and StreetDance(e). The best cluster trees empirically selected for Synthetic-skin1 ($\beta = 0.99$), Martin-skin2 ($\beta = 0.95$), DisneyFace ($\beta = 0.98$), Garment ($\beta = 0.994$) and StreetDance ($\beta = 0.996$).

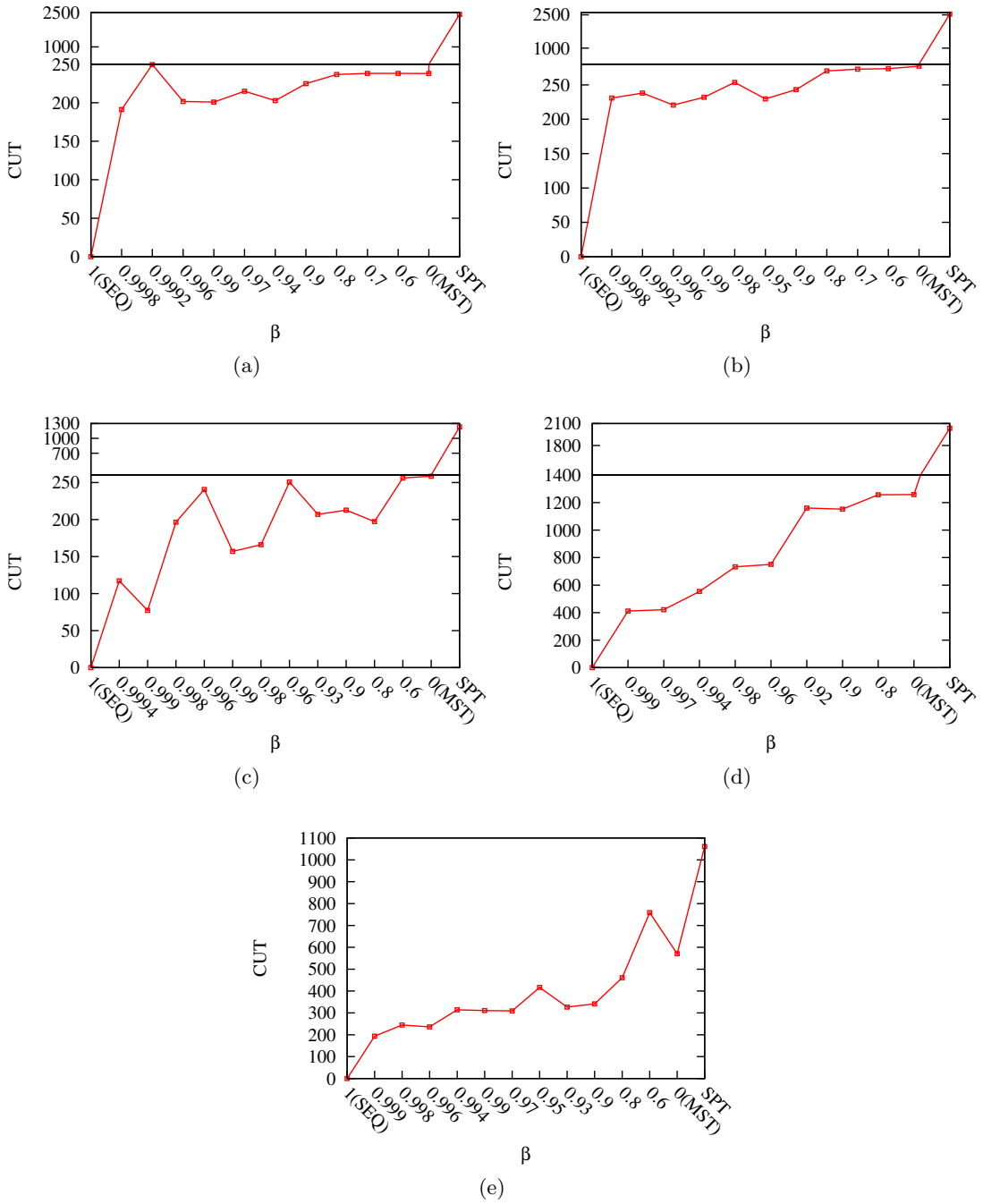


Figure D.3: *CUT* measure across different traversals for the datasets Synthetic-skin1(a), Martin-skin2(b), DisneyFace(c), Garment(d) and StreetDance(e). The best cluster trees empirically selected for Synthetic-skin1 ($\beta = 0.99$), Martin-skin2 ($\beta = 0.95$), DisneyFace ($\beta = 0.98$), Garment ($\beta = 0.994$) and StreetDance ($\beta = 0.996$).

Appendix E

Facial performance capture - additional results

Example frames from the temporally consistent 3D model of the face for the datasets Martin-makeup1 and Alaleh-makeup1 are depicted in Figures E.1 and E.2. Large figures showcase the amount of geometric detail obtained by the proposed capture system.

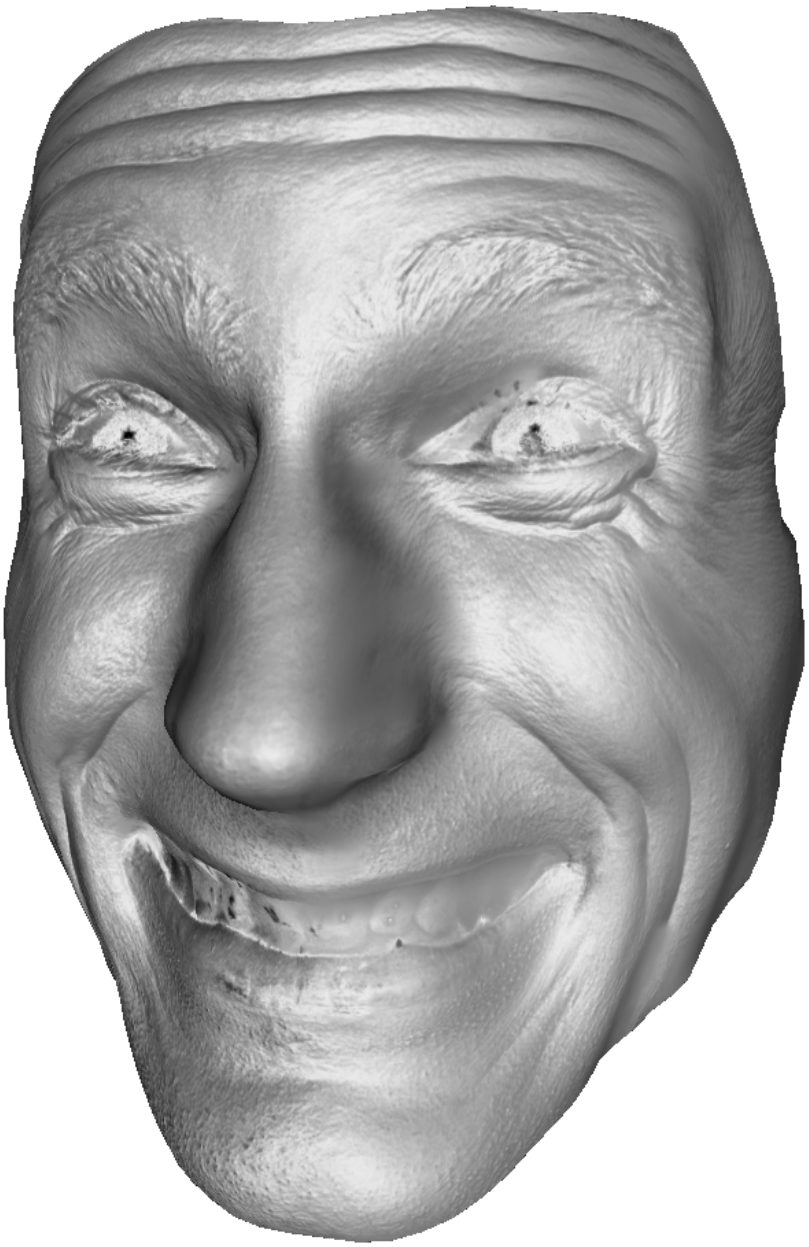


Figure E.1: The temporally consistent 3D model of the face at the frame 120 from the dataset Martin-makeup1.

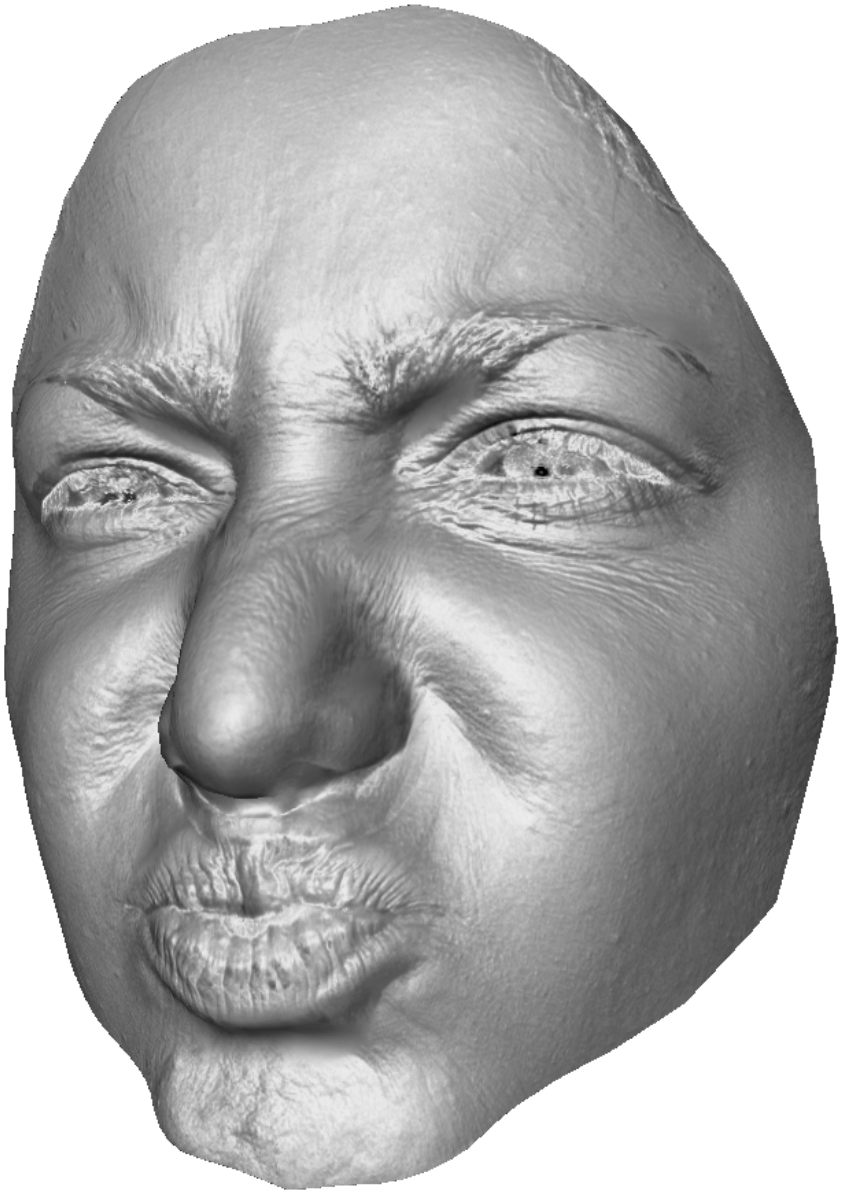


Figure E.2: The temporally consistent 3D model of the face at the frame 111 from the dataset Alaleh-makeup1.

Appendix F

Computation time

Analysis of computation time is performed for individual algorithmic tasks in the pipeline of the proposed system for facial performance capture (Chapter 7). The test dataset is a performance Martin-makeup1 with 300 frames (details in Appendix G). Algorithmic implementation is single-threaded C++ code which is processed on Intel CORE i5 processor (3.3GHz). Table F.1 lists the tasks computed for each frame of the performance and their contributions to per-frame computation time of $1.80min$. Table F.2 lists the tasks performed once for the whole sequence and the total contribution of per-frame computation. This amounts to the overall computation time of $838.13min$ for the dataset Martin-makeup1.

Certain tasks can be performed in parallel because they are view-dependent. This is reflected by the columns Total time and Effective time in Tables F.1,F.2 where the total time sums all processing across views and the effective time is the actual time spent with tasks running in parallel if possible. Note that non-sequential tracking makes one frame-to-frame alignment per frame only if the multi-path temporal fusion across cuts is not used. With the temporal fusion, the overall number of alignments is higher because some frames are tracked multiple times (643 alignments in this case). Thus, the temporal alignment contribution per frame in Table F.1 is not entirely correct but the computation time for non-sequential tracking is separated in Table F.2 to highlight the difference. The time with the temporal fusion is added to the total time for the whole sequence.

Processing of a performance sequence involves prior steps which need some user interaction. The capture setup requires camera calibration and calibration for photometric stereo. A coarse version of the facial mesh which is tracked needs to be designed by the user. Also, linear predictor tracker necessary for dissimilarity computation requires manual landmarking of sparse set of points in several frames.

Task	Total time	Effective time
Stereo matching	$0.90min$	$0.45min$
Poisson reconstruction	$0.32min$	$0.16min$
Frame-to-frame alignment	$0.82min$	$0.82min$
PSCL	$0.08min$	$0.04min$
Normal correction	$0.50min$	$0.25min$
Normal mapping	$0.08min$	$0.08min$
Total	$2.70min$	$1.80min$

Table F.1: Computational time for individual tasks per frame.

Task	Total time	Effective time
Linear predictor training	$20.00min$	$10.00min$
Linear predictor tracking	$9.76min$	$4.88min$
Dissimilarity computation	$0.20min$	$0.20min$
Frame clustering	$1.02min$	$1.02min$
Cluster tree calculation	$0.02min$	$0.02min$
Non-seq. tracking (without temporal fusion)	$246.35min$	$246.35min$
Non-seq. tracking (with temporal fusion)	$528.01min$	$528.01min$
Other per-frame computations	$564.00min$	$294.00min$
Total	$1123.01min$	$838.13min$

Table F.2: Computational time for individual tasks over the whole sequence Martin-makeup (300 frames). The first block are the tasks performed once for the sequence. The second block contains aggregated times from per-frame computations.

Appendix G

Datasets

Technical description of datasets used for evaluations throughout this thesis is provided. Table G.1 lists individual datasets with information about multi-view image sequences captured. Camera calibration data and per-frame 3D reconstructions of an actor are also available for all datasets.

The datasets Martin-pattern1, Martin-markers1, Martin-skin1, Martin-skin2 and Garment were acquired under white diffuse illumination in HD-SDI uncompressed $4:2:2$ format by the capture setup described in Section 7.2. The datasets Martin-makeup1 and Alaleh-makeup1 were captured under directional colour illumination in HD-SDI uncompressed $4:4:4$ format by the same capture setup. The dataset Synthetic-skin1 is a synthetic facial performance rendered into views of a similar virtual capture setup.

The dataset DisneyFace has been released by ETH/Disney Research Zurich [13] and the facial performance was recorded by 7 cameras under white diffuse illumination. The dataset StreetDance is publicly available thanks to Centre for Vision, Speech and Signal Processing at University of Surrey [101]. A studio setup with 8 HD cameras surrounding an actor was used. This dataset is a concatenation of 3 separate full-body performances by a breakdancer - Free, KickUp, FlashKick.

Dataset	No. of cameras	Resolution	Fps	No. of frames	Frame range
Synthetic-skin1	4	800×950	25	355	0 - 354
Martin-pattern1	4	1920×1080	25	310	1 - 310
Martin-markers1	4	1920×1080	25	367	70 - 436
Martin-skin1	4	1920×1080	25	310	0 - 309
Martin-skin2	4	1920×1080	25	355	0 - 354
Martin-makeup1	4	1920×1080	25	300	0 - 299
Martin-makeup2	4	1920×1080	25	250	0 - 249
Alaleh-makeup1	4	1920×1080	25	341	42 - 382
DisneyFace	7	1176×864	46	347	48 - 394
Garment	4	1920×1080	25	320	310 - 629
Free				500	
KickUp				300	
FlashKick				250	
StreetDance	8	1920×1080	25	1050	0 - 1049

Table G.1: Description of the datasets used for evaluations.

Bibliography

[1] 3dMD. 3dMDdynamic System. `http://www.3dmd.com/3dmd4d-capture.html`.

[2] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec. Creating a Photoreal Digital Actor: The Digital Emily Project. In *Proceedings of the European Conference on Visual Media Production*, pages 176–187, 2009.

[3] R. Anderson and B. Stenger. Augmenting depth camera output using photometric stereo. *Machine Vision and Applications*, 2011.

[4] R. Anderson, B. Stenger, and R. Cipolla. Color Photometric Stereo for Multicolored Surfaces. In *Proceedings of the International Conference on Computer Vision*, pages 2182–2189, 2011.

[5] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(1):221–255, 2004.

[6] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics*, 28(3), 2009.

[7] C. Barnes, E. Shechtman, D. Goldman, and A. Finkelstein. The Generalized PatchMatch Correspondence Algorithm. In *Proceedings of the European Conference on Computer Vision*, pages 29–43, 2010.

[8] S. Barsky and M. Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1239–1252, 2003.

[9] S. Barsky and M. Petrou. Design issues for a colour photometric stereo system. *Journal of Mathematical Imaging and Vision*, 24(1):143–162, 2006.

[10] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-Quality Single-Shot Capture of Facial Geometry. *ACM Transactions on Graphics*, 29(4):1, 2010.

[11] T. Beeler, B. Bickel, G. Noris, P. Beardsley, S. Marschner, Robert W. Sumner, and M. Gross. Coupled 3D reconstruction of sparse facial hair and skin. *ACM Transactions on Graphics*, 31(4):1–10, 2012.

[12] T. Beeler, D. Bradley, H. Zimmer, and M. Gross. Improved Reconstruction of Deforming Surfaces by Cancelling Ambient Occlusion. In *Proceedings of the European Conference on Computer Vision*, pages 30–43, 2012.

[13] T. Beeler, F. Hahn, D. Bradley, and B. Bickel. High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics*, 30(212):1–10, 2011.

[14] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H.P. Pfister, and M. Gross. Multi-scale capture of facial geometry and motion. *ACM Transactions on Graphics*, 26(3):33, 2007.

[15] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the Annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[16] Blender Foundation. Blender v2.59. `http://www.blender.org`.

[17] G. Borshukov, D. Piponi, O. Larsen, J. P. Lewis, and C. Tempelaar-Lietz. Universal capture: image-based facial animation for "The Matrix Reloaded". In *Proceedings of the ACM SIGGRAPH Sketches & Applications*, pages 1–1, 2003.

[18] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[19] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High Resolution Passive Facial Performance Capture. *ACM Transactions on Graphics*, 29(4):41, 2010.

[20] G. Brostow, C. Hernandez, G. Vogiatzis, B. Stenger, and R. Cipolla. Video normals from colored lights. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (i):1–12, 2011.

[21] A. Buchanan and A. Fitzgibbon. Interactive Feature Tracking using K-D Trees and Dynamic Programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 626–633, 2006.

[22] C. Budd, P. Huang, and A. Hilton. Hierarchical Shape Matching for Temporally Consistent 3D Video. In *Proceedings of the International Conference on 3D Imaging, Modelling, Processing, Visualization and Transmission*, pages 172–179, 2011.

[23] R.L. Carceroni and K.N. Kutulakos. Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape and reflectance. *International Journal of Computer Vision*, 49(2-3):175–214, 2002.

[24] M. Chandraker, S. Agarwal, and D. Kriegman. ShadowCuts: Photometric Stereo with Shadows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[25] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[26] J. Courchay, J.-P. Pons, P. Monasse, and R. Keriven. Dense and Accurate Spatio-Temporal Multi-View Stereovision. In *Proceedings of the Asian Conference on Computer Vision*, pages 11–22, 2009.

[27] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the Annual conference on Computer graphics and interactive techniques*, pages 145–156, 2000.

[28] D. DeCarlo. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 231–238, 1996.

[29] F. Devernay, D. Mateus, and M. Guilbert. Multi-Camera Scene Flow by Tracking 3-D Points and Surfels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2203–2212, 2006.

[30] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[31] Dimensional Imaging. DI4D - 4D Capture Systems. `http://www.di3d.com/products/4d_tracking/`.

[32] O. Drbohlav and M. Chantler. On optimal light configurations in photometric stereo. In *Proceedings of the International Conference on Computer Vision*, pages 1707–1712, 2005.

[33] M. Drew. Shape from color. Technical Report Technical Report CSS/LCCR TR 92-07, Simon Fraser University School of Computing Science, 1992. `ftp://fas.sfu.ca/pub/cs/techreports/1992/CSS-LCCR92-07.ps.gz`.

[34] I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, 2002.

[35] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, 1978.

[36] Y. Furukawa and J. Ponce. Dense 3d motion capture from synchronized video streams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[37] Y. Furukawa and J. Ponce. Dense 3d motion capture for human faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1674–1681, 2009.

[38] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[39] G. Fyffe, T. Hawkins, C. Watts, W.-C. Ma, and P. Debevec. Comprehensive Facial Performance Capture. *Computer Graphics Forum*, 30(2):425–434, 2011.

[40] G. Fyffe and X. Yu. Single-shot photometric stereo by spectral multiplexing. In *Proceedings of the IEEE International Conference on Computational Photography*, pages 1–6, 2011.

[41] A. Ghosh, G. Fyffe, and B. Tunwattanapong. Multiview face capture using polarized spherical gradient illumination. *ACM Transactions on Graphics*, 30(6):1–10, 2011.

[42] M. Glencross, G. Ward, F. Melendez, C. Jay, J. Liu, and R. Hubbold. A perceptually validated model for surface depth hallucination. *ACM Transactions on Graphics*, 27(3):1, 2008.

[43] D. Goldman, B. Curless, A. Hertzmann, and S. Seitz. Shape and spatially-varying BRDFs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–71, 2010.

[44] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *Proceedings of the Annual conference on Computer graphics and interactive techniques*, pages 55–66, 1998.

[45] T. Hawkins, A. Wenger, C. Tchou, A. Gardner, F. Goransson, and P. Debevec. Animatable facial reflectance fields. In *Proceedings of the Eurographics Symposium on Rendering*, pages 309–319, 2004.

[46] C. Hernández. Self-calibrating a real-time monocular 3d facial capture system. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission*, 2010.

[47] C. Hernández, G. Vogiatzis, G. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007.

[48] C. Hernández, G. Vogiatzis, and R. Cipolla. Shadows in three-source photometric stereo. In *Proceedings of the European Conference on Computer Vision*, pages 290–303, 2008.

[49] C. Hernández, G. Vogiatzis, and R. Cipolla. Overcoming shadows in 3-source photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):419–426, 2011.

[50] A. Hertzmann and S. Seitz. Example-based photometric stereo: shape reconstruction with general, varying BRDFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1254–64, 2005.

[51] H. Huang, J. Chai, X. Tong, and H.-T. Wu. Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition. *ACM Transactions on Graphics*, 30(4):1, 2011.

[52] P. Huang, C. Budd, and A. Hilton. Global temporal registration of multiple non-rigid surface sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3473–3480, 2011.

[53] P. Huang, A. Hilton, and J. Starck. Automatic 3d video summarization: Key frame extraction from self-similarity. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.

[54] P. Huang, A. Hilton, and J. Starck. Shape Similarity for 3D Video Sequences of People. *International Journal of Computer Vision*, 89(2-3):362–381, 2010.

[55] X. Huang, S. Zhang, Y. Wang, D. Metaxas, and D. Samaras. A Hierarchical Framework For High Resolution Facial Expression Tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, pages 22–22, 2004.

[56] F. Huguet and F. Devernay. A Variational Method for Scene Flow Estimation from Stereo Sequences. In *Proceedings of the International Conference on Computer Vision*, pages 1–7, 2007.

[57] Image Metrics. Emily Project. `http://gl.ict.usc.edu/Research/DigitalEmily/`.

[58] X. Jiang and H. Bunke. On error analysis for surface normals determined by photometric stereo. *Signal Processing*, 23(3):221–226, 1991.

[59] A. Jones, A. Gardner, M. Bolas, I. McDowall, and P. Debevec. Simulating spatially varying lighting on a live performance. In *Proceedings of the European Conference on Visual Media Production*, pages 127–133, 2006.

[60] M. Kazhdan and M. Bolitho. Poisson surface reconstruction. In *Proceedings of the Eurographics Symposium on Geometry Processing*, pages 61–70, 2006.

[61] Khronos Group. Open GL. `http://www.opengl.org/`.

[62] H. Kim, B. Wilburn, and M. Ben-Ezra. Photometric stereo for dynamic surface orientations. In *Proceedings of the European Conference on Computer Vision*, pages 59–72, 2010.

[63] M. Langer and H. Bülthoff. Depth discrimination from shading under diffuse lighting. *Perception*, 29(6):649–660, 2000.

[64] H. Li, B. Adams, L. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics*, 28(5):1, 2009.

[65] R. Li and S. Sclaroff. Multi-scale 3d scene flow from binocular stereo sequences. *Computer Vision and Image Understanding*, 110(1):75–90, 2008.

[66] I.-C. Lin and M. Ouhyoung. Mirror MoCap: Automatic and efficient capture of dense 3D facial motion parameters from video. *The Visual Computer*, 21(6):355–372, 2005.

[67] W.-C. Ma, T. Hawkins, P. Peers, C.-F. Chabert, M. Weiss, and P. Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Proceedings of the Eurographics Symposium on Rendering*, pages 183–194, 2007.

[68] W.-C. Ma, A. Jones, J.-Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, and P. Debevec. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Transactions on Graphics*, 27(5):1–10, 2008.

[69] W.-C. Ma, A. Jones, T. Hawkins, J.-Y. Chiang, and P. Debevec. A high-resolution geometry capture system for facial performance. In *Proceedings of the ACM SIGGRAPH talks*, number 3, 2008.

[70] S. Marschner, B. Guenter, and S. Raghupathy. Modeling and rendering for realistic facial animation. In *Proceedings of the Eurographics Symposium on Rendering*, pages 1–13, 2000.

[71] M. Meyer, M. Desbrun, P. Schroder, and A. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics*, III:35–57, 2003.

[72] Microsoft. Kinect. `http://www.xbox.com/en-GB/Kinect`.

[73] D. Min and K. Sohn. Edge-preserving Simultaneous Joint Motion-Disparity Estimation. In *Proceedings of the International Conference on Pattern Recognition*, pages 74–77, 2006.

[74] J. Mitchelson. Wand-based Multiple Camera Studio Calibration. Technical report, Centre for Vision, Speech, and Signal Processing, University of Surrey, UK, 2003.

[75] M. Mori. The Uncanny Valley. In *Energy*, pages 33–35, 1970.

[76] Mova. CONTOUR Reality Capture. `http://www.mova.com/technology.php`.

[77] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. *ACM Transactions on Graphics*, 24(3):536–543, 2005.

[78] J. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[79] J. Neumann and Y. Aloimonos. Spatio-Temporal Stereo Using Multi-Resolution Subdivision Surfaces. *International Journal of Computer Vision*, 47(1-3):181–193, 2002.

[80] Nvidia. Cg shading language. `https://developer.nvidia.com/cg-toolkit`.

[81] E.-J. Ong, Y. Lan, B. Theobald, R. Harvey, and R. Bowden. Robust Facial Feature Tracking using Selected Multi-Resolution Linear Predictors. In *Proceedings of the International Conference on Computer Vision*, pages 1483–1490, 2009.

[82] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, 2006.

[83] A. Patel and W. Smith. Driving 3D morphable models using shading cues. *Pattern Recognition*, 45(5):1993–2004, 2012.

[84] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and David H. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of the Annual conference on Computer graphics and interactive techniques*, pages 75–84, 1998.

[85] F. Pighin, R. Szeliski, and D. Salesin. Resynthesizing facial animation through 3D model-based tracking. In *Proceedings of the International Conference on Computer Vision*, pages 143–150, 1999.

[86] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193, 2007.

[87] T. Popa, I. South-Dickinson, D. Bradley, A. Sheffer, and W. Heidrich. Globally Consistent Space-Time Reconstruction. *Computer Graphics Forum*, 29(5):1633–1642, 2010.

[88] R. Prim. Shortest connection networks and some generalizations. *Bell System Technology Journal*, 36:1389–1401, 1957.

[89] R. Ray, J. Birk, and R. Kelley. Error analysis of surface normals determined by radiometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(6):631–645, 1983.

[90] S. Roy. Stereo without epipolar lines: A maximum-flow formulation. *International Journal of Computer Vision*, 34(2/3):147–161, 1999.

[91] P. Sand and S. Teller. Particle Video: Long-Range Motion Estimation Using Point Trajectories. *International Journal of Computer Vision*, 80(1):72–91, 2008.

[92] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006.

[93] D. Sibbing, M. Habbecke, and L. Kobbelt. Markerless Reconstruction and Synthesis of Dynamic Facial Expressions. *Computer Vision and Image Understanding*, 115(5), 2011.

[94] M. Smith and L. Smith. Dynamic photometric stereo–a new technique for moving surface analysis. *Image and Vision Computing*, 23(9):841 – 852, 2005.

[95] W. Smith and E. Hancock. Recovering facial shape using a statistical model of surface normal direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1914–30, 2006.

[96] W. Smith and E. Hancock. A new framework for grayscale and colour non-lambertian shape-from-shading. In *Proceedings of the Asian Conference on Computer Vision*, pages 869–880, 2007.

[97] W. Smith and E. Hancock. Estimating Facial Reflectance Properties Using Shape-from-Shading. *International Journal of Computer Vision*, 86(2-3):152–170, 2010.

[98] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the Eurographics Symposium on Geometry Processing*, pages 109–116, 2007.

[99] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the Eurographics Symposium on Geometry Processing*, pages 175–184, 2004.

[100] A. Spence and M. Chantler. Optimal illumination for three-image photometric stereo using sensitivity analysis. *IEE Proceedings - Vision, Image and Signal Processing*, 153(2):149–159, 2006.

[101] J. Starck and A. Hilton. Surface capture for performance-based animation. *Computer Graphics and Applications*, 27(3):21–31, 2007.

[102] J. Sun, M. Smith, L. Smith, and A. Farooq. Examining the uncertainty of the recovered surface normal in three light photometric stereo. *Image and Vision Computing*, 25(7):1073–1079, 2007.

[103] The Foundry. Nuke v6.0. `http://www.thefoundry.co.uk/products/nuke/`.

[104] S. Vedula, S. Baker, and T. Kanade. Image-based spatio-temporal modeling and view interpolation of dynamic events. *ACM Transactions on Graphics*, 24(1):240 – 261, 2005.

[105] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the International Conference on Computer Vision*, pages 722–729, 1999.

[106] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):475–80, 2005.

[107] VICON. VICON Cara. `http://www.vicon.com/products/cara.html`.

[108] D. Vlasic, M. Brand, H.-P. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Transactions on Graphics*, 24(3):426–433, 2005.

[109] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Transactions on Graphics*, 28(5):1, 2009.

[110] G. Vogiatzis and C. Hernández. Self-calibrated, Multi-spectral Photometric Stereo for 3D Face Capture. *International Journal of Computer Vision*, 97(1):91–103, 2012.

[111] C. Walder, M. Breidt, H. Bülthoff, and B. Schölkopf. Markerless 3d face tracking. In *Proceedings of the DAGM Symposium for Pattern Recognition*, pages 41–50, 2009.

[112] Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and P. Huang. High Resolution Tracking of Non-Rigid Motion of Densely Sampled 3D Data Using Harmonic Maps. *International Journal of Computer Vision*, 76(3):283–300, 2008.

[113] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic Scene Flow Computation for 3D Motion Understanding. *International Journal of Computer Vision*, 95(1):29–51, 2010.

[114] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics*, 30(4):1, 2011.

[115] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/Off: Live Facial Puppetry. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 7–16, 2009.

[116] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics*, 24(3):756–764, 2005.

[117] T. Weyrich, W. Matusik, H. Pfister, and B. Bickel. Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics*, 25(3):1013–1024, 2006.

[118] L. Williams. Performance-driven facial animation. In *Proceedings of the Annual conference on Computer graphics and interactive techniques*, pages 235 – 242, 1990.

[119] C. Wilson, A. Ghosh, P. Peers, J.-Y. Chiang, J. Busch, and P. Debevec. Temporal upsampling of performance geometry using photometric alignment. *ACM Transactions on Graphics*, 29(2):1–11, 2010.

[120] R. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139 – 44, 1980.

[121] I. Ypsilos, A. Hilton, and S. Rowe. Video-rate capture of dynamic face shape and appearance. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 117–122, 2004.

[122] L. Zhang, N. Snavely, B. Curless, and S. Seitz. Spacetime faces: High-resolution capture for modeling and animation. *ACM Transactions on Graphics*, 23:548–558, 2004.

[123] R. Zhang, P. Tsai, J. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

[124] S. Zhang and P. Huang. High-resolution, real-time three-dimensional shape measurement. *Optical Engineering*, 45(12):123601, 2006.

[125] Y. Zhang and C. Kambhamettu. Integrated 3D scene flow and structure recovery from multiview image sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 674–681, 2000.

[126] Y. Zhang and C. Kambhamettu. On 3D scene flow and structure estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 778–785, 2001.

[127] Y. Zhang and C. Kambhamettu. On 3-D scene flow and structure recovery from multiview image sequences. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(4):592–606, 2003.

[128] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[129] W. Zhou and C. Kambhamettu. Estimation of illuminant direction and intensity of multiple light sources. In *Proceedings of the European Conference on Computer Vision*, pages 206–220, 2002.