Northumbria University
NEWCASTLE

METRIC REPRESENTATIONS FOR SHAPE ANALYSIS AND SYNTHESIS

SHANFENG HU

PhD

2020

METRIC REPRESENTATIONS FOR SHAPE ANALYSIS AND SYNTHESIS

SHANFENG HU

A thesis submitted in partial fulfilment of the requirements of the University of Northumbria at Newcastle for the degree of Doctor of Philosophy

Faculty of Engineering and Environment

March 2020

Abstract

2D and 3D geometric shapes are ubiquitous in computer graphics, computer animation, and computer-aided design and manufacturing. Two of the fundamental research challenges that underline these applications are the analysis and synthesis of shapes, with the former aiming to extract semantically meaningful knowledge of shapes and the latter focusing on generating plausible-looking shapes based on user inputs. Traditionally, shape analysis and synthesis are based on representations such as meshes, parameterisations, and Laplacians, which lead to mostly hand-crafted computation rules that are either suboptimal or treat related tasks separately. In this work, we propose to represent a 2D/3D shape as a square symmetric matrix that correlates every pair of geometric points on the shape, which allows us to formulate shape analysis and synthesis problems as principled optimisation problems that can be globally optimised. To demonstrate the usefulness of our new metric representation for shape analysis, we first address 3D mesh saliency detection by representing a shape as a pairwise feature distance matrix, whose principal eigenvector is experimentally shown to outperform the traditional saliency detection rules for capturing ground-truth saliency annotations. Following this work, we then unify saliency detection and non-rigid shape matching via a jointly learned metric representation, which is shown to improve the accuracy of both tasks on the existing saliency detection and shape matching benchmarks. To also demonstrate the usefulness of our metric representation for shape synthesis, we address 2D facial shape beautification in images by representing a facial shape as the orthogonal projection matrix onto 2D facial landmarks, which is shown to improve the attractiveness of both frontal-neutral and non-frontal-non-neutral faces in the user studies. Finally, we show that adversarially learning the distributions of human shapes and poses in a hidden space produces higher-quality human samples than in the geometry space. Together, these results show that our metric representation benefits both the analysis and synthesis of shapes, with the potential of unifying more diverse tasks such as part segmentation and labelling in the future work.

Keywords: shape analysis, shape synthesis, face beautification, mesh saliency detection, non-rigid shape matching, generative modelling, pose, motion, metric, sparsity, deformation invariance, orthogonal projection metric, Laplacian, eigenvectors, probability distribution, deep learning, recurrent neural networks, generative adversarial networks, optimisation

Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two-dimensional
3D	Three-dimensional
BIM	Blended Intrinsic Maps
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
DFM	Deep Functional Maps
DSPP	Deep Shape and Pose Priors of Humans
Eigengap	Gap of Eigenvalues
FAUST	Dataset and Evaluation for 3D Mesh Registration
GAN	Generative Adversarial Network
GANs	Generative Adversarial Networks
GRU	Gated Recurrent Unit
HKCNN	Heat Kernel Convolutional Neural Network
HKS	Heat Kernel Signature
IL	Inner Lip
JL	Jawline
KD-tree	K-dimensional Tree
LB	Left Eyebrow
LE	Left Eye
LSTM	Long-short Term Memory
MLP	Multi-layer Perceptron

MLP-SAL	Multi-layer Perceptron with Saliency
MPII	Max Planck Institute for Informatics
MR	Manifold Ranking
MS	Mesh Saliency
NB	Nose Bridge
NO	Nostril
OL	Outer Lip
PCA	Principal Component Analysis
PhD	Doctor of Philosophy
PointNet	Point Neural Network
PointNet++	Point Neural Network Plus Plus
PReLU	Parametric Rectified Linear Unit
RB	Right Eyebrow
RE	Right Eye
Ref.	Reference
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
RNNs	Recurrent Neural Networks
s.t.	Subject To
SCAPE	Shape Completion and Animation of People
SDC	Scale-dependent Corners
SDP	Semi-definite Programming
SDP-SAL	Semi-definite Programming with Saliency
SFU	Simon Fraser University
SH	Spherical Harmonic
SHREC	Shape Retrieval Contest
SHREC2007	Shape Retrieval Contest 2007

SI	Spectral Irregularity
sigmoid	Sigmoid Function
SMPL	Skinned Multi-Person Linear Model
SN	Spectral Normalization
SP	Sparse Points
SRI	Surface Regions of Interest
ST	Soft-thresholding
SurfCNN	Surface Convolutional Neural Network
tanh	Hyperbolic Tangent Function
TBR	Tree-based Regression
Trans.	Transformation
UK	United Kingdom
VAE	Variational Auto-encoding
WKS	Wave Kernel Signature

List of Symbols

$\mathbb{R}$	The field of real numbers
$\mathbb{R}_{\geq 0}$	The field of non-negative real numbers
$\boldsymbol{I}$	The identity matrix
$I[\cdot]$	The indicator function
$\mathbf{1}$	A vector of all ones
$\mathbf{0}$	A vector of all zeros
$\mathcal{N}(0, \boldsymbol{I})$	The standard normal distribution
log$\cdot$	The log function
$\perp$	Orthogonal
$\arg\max$	Maximisation problem
$\arg\min$	Minimisation problem
∞	The infinity
$\|\cdot\|, \|\cdot\|_2$	The Euclidean (L2) norm of a vector
$\|\cdot\|^2, \|\cdot\|_2^2$	The Squared Euclidean (L2) norm of a vector
$\|\cdot\|_F$	The Frobenius norm of a matrix
$\|\cdot\|_F^2$	The squared Frobenius norm of a matrix
$\|\cdot\|_0$	The L0 norm (i.e. number of non-zero elements) of a vector
$\|\cdot\|$	The element-wise absolute values of input
i, j, k	The index of points
τ	The index of scales
N	The number of points

N_τ	The number of scales
$\mathcal{P}$	The point set of a surface mesh
$\mathcal{S}$	The set of scales
$\boldsymbol{p}^i$	A point from a point set
$\boldsymbol{p}^{i,\tau}$	A spherical region centred at point $\boldsymbol{p}^i$ with scale τ
$\boldsymbol{f}^{i,\tau}$	The shape descriptor of a spherical region centred at point $\boldsymbol{p}^i$ with scale τ
$\boldsymbol{M}(\mathcal{P})$	The feature distance metric of a mesh
$\boldsymbol{M}^\tau(\mathcal{P})$	The feature distance metric of a mesh at scale τ
$\boldsymbol{M}_{ij}(\mathcal{P})$	The feature distance between point $\boldsymbol{p}^i$ and $\boldsymbol{p}^j$
$\boldsymbol{M}_{ij}^\tau(\mathcal{P})$	The feature distance between point $\boldsymbol{p}^i$ and $\boldsymbol{p}^j$ at scale τ
$\boldsymbol{\varphi}(\mathcal{P})$	The saliency map of a mesh
$\boldsymbol{\varphi}^\tau(\mathcal{P})$	The saliency map of a mesh at scale τ
$\boldsymbol{\varphi}_i(\mathcal{P})$	The saliency value of point $\boldsymbol{p}^i$
$\boldsymbol{\varphi}_i^\tau(\mathcal{P})$	The saliency value of point $\boldsymbol{p}^i$ at scale τ
$\boldsymbol{\varphi}_{(i)}(\mathcal{P})$	The non-decreasing saliency value of point $\boldsymbol{p}^i$
$\bar{\boldsymbol{\varphi}}(\mathcal{P})$	The ground-truth saliency map of a mesh
$\boldsymbol{E}(\mathcal{P})$	The shape embeddings of a mesh
$\boldsymbol{E}_{i\cdot}(\mathcal{P})$	The shape embeddings of point $\boldsymbol{p}^i$
$R(\boldsymbol{\varphi}(\mathcal{P}))$	The rarity objective value of the saliency map of a mesh
μ	The fraction of salient points between 0 and 1
σ	The scale of a Gaussian filter kernel
θ	A convex combination coefficient between 0 and 1
$\lambda_i(\cdot)$	The non-decreasing eigenvalues of a matrix
$\boldsymbol{C}(\mathcal{P})$	The cotangent-weight affinity matrix of a mesh
$\boldsymbol{S}(\mathcal{P})$	The salient affinity matrix of a mesh
$\boldsymbol{A}(\mathcal{P})$	The affinity matrix of a mesh
$\boldsymbol{\Delta}(\cdot)$	The Laplacian operator of a matrix
$\mathrm{tr}(\cdot)$	The trace of a matrix

$\boldsymbol{W}^{\diamond,l}, \boldsymbol{M}^{\diamond,l}$	The weight parameters of an RNN at layer l
$\boldsymbol{F}^{\tau,l}(\mathcal{P})$	The feature matrix of a mesh at scale τ at layer l
$\boldsymbol{\Upsilon}(\cdot)$	The feature standardisation operator
$\odot$	The element-wise product
$\max\{\cdot,\cdot\}$	The element-wise maximum operator
Θ_t	The soft-thresholding parameter
$\mathcal{L}(\mathcal{P},\mathcal{P}')$	The unified loss function for a pair of meshes
$\mathcal{L}_\alpha(\mathcal{P})$	The saliency fitting loss for a mesh
$\mathcal{L}_\beta(\mathcal{P},\mathcal{P}')$	The saliency consistency loss for a pair of meshes
$\mathcal{L}_\gamma(\mathcal{P},\mathcal{P}')$	The metric consistency loss for a pair of meshes
α,β,γ	The weights for the saliency fitting, saliency consistency, and metric consistency losses respectively
$\boldsymbol{M}$	The orthogonal projection metric of a facial shape
$\boldsymbol{M}_{ij}$	An element of the orthogonal projection metric of a facial shape
$\boldsymbol{M}^*$	The beautified orthogonal projection metric of a facial shape
$\boldsymbol{M}^*_{ij}$	An element of the beautified orthogonal projection metric of a facial shape
$\overline{\boldsymbol{M}}$	The mean of facial metrics
$\boldsymbol{P}$	The homogeneous landmark coordinates of a facial shape
$\boldsymbol{P}^*$	The beautified homogeneous landmark coordinates of a facial shape
$\boldsymbol{P}_{i1}, \boldsymbol{P}_{i2}, \boldsymbol{P}_{i3}$	The homogeneous coordinates of a facial landmark
$\overline{\boldsymbol{P}}$	The mean of facial shapes
$\boldsymbol{L}$	A linear transformation matrix
$\boldsymbol{T}$	A translation vector
$\boldsymbol{S}$	The covariance matrix of the homogeneous landmarks of a facial shape
$\mathcal{M}$	A set of facial metrics
$\boldsymbol{\Gamma}_c$	A principal component
λ_c	The corresponding eigenvalue of a principal component
C	The number of principal components

$\Psi(\cdot)$	The mean-shift local averaging operator
h_i	The adaptive mean-shift bandwidth of a face example in a dataset
$\boldsymbol{w}$	The vector of user-specified beautification weights for each landmark
$\boldsymbol{W}$	The matrix of user-specified beautification weights for the whole face
$\boldsymbol{p}(x)$	The empirical probability distribution of realistic human shapes
$\hat{\boldsymbol{p}}(x)$	The estimated probability distribution of realistic human shapes
$x \sim \boldsymbol{p}(x)$	Human shape synthesis (sampling) in the geometry space
$\boldsymbol{p}(z_x)$	The empirical probability distribution of human shape auto-encoding features
$\hat{\boldsymbol{p}}(z_x)$	The approximate probability distribution of human shape auto-encoding features
$z_x \sim \boldsymbol{p}(z_x)$	Human shape synthesis (sampling) in the auto-encoding space
$\boldsymbol{p}(y)$	The empirical probability distribution of realistic human poses
$\hat{\boldsymbol{p}}(y)$	The estimated probability distribution of realistic human poses
$y \sim \boldsymbol{p}(y)$	Human pose synthesis (sampling) in the geometry space
$\boldsymbol{p}(z_y)$	The empirical probability distribution of human pose auto-encoding features
$\hat{\boldsymbol{p}}(z_y)$	The approximate probability distribution of human pose auto-encoding features
$z_y \sim \boldsymbol{p}(z_y)$	Human pose synthesis (sampling) in the auto-encoding space
f_{shape}	The encoder for human shapes
h_{shape}	The decoder for human shapes
f_{pose}	The encoder for human poses
h_{pose}	The decoder for human poses
$l(x, f_{\text{shape}}, h_{\text{shape}})$	The auto-encoder reconstruction loss for human shapes
$l(y, f_{\text{pose}}, h_{\text{pose}})$	The auto-encoder reconstruction loss for human poses
$\varphi_{\text{shape}} : z \to z_x$	The generator for human shapes
$\psi_{\text{shape}} : z_x \to (0, 1)$	The discriminator for human shapes
$\varphi_{\text{pose}} : z \to z_y$	The generator for human poses
$\psi_{\text{pose}} : z_y \to (0, 1)$	The discriminator for human poses

$l(\hat{\boldsymbol{p}}(z_x), \boldsymbol{p}(z_x))$	The generative adversarial loss for human shapes
$l(\hat{\boldsymbol{p}}(z_y), \boldsymbol{p}(z_y))$	The generative adversarial loss for human poses

Publications

- **Shanfeng Hu**, Hubert P. H. Shum, Nauman, Aslam, Frederick W. B. Li and Xiaohui Liang (2020), 'A Unified Deep Metric Representation for Mesh Saliency Detection and Non-rigid Shape Matching', IEEE Transactions on Multimedia.
- **Shanfeng Hu**, Xiaohui Liang, Hubert P. H. Shum, Frederick W. B. Li and Nauman Aslam (2020), 'Sparse Metric-based Mesh Saliency', Neurocomputing.
- **Shanfeng Hu**, Hubert P. H. Shum, Xiaohui Liang, Frederick W. B. Li and Nauman Aslam (2020), 'Facial Reshaping Operator for Controllable Face Beautification', Expert Systems with Applications (minor revision submitted).
- **Shanfeng Hu**, Hubert P. H. Shum, Antonio Mucherino (2019), 'DSPP: Deep Shape and Pose Priors of Humans', ACM SIGGRAPH International Conference on Motion, Interaction, and Games.

Acknowledgements

Completing the PhD study has been and will be one of the most rewarding life journeys of mine. This journey would not be possible without the continuous support from my parents, who did not receive much school education when they were young but have always been financing and encouraging me to pursue knowledge, from high school to university and finally to graduate school. The words "knowledge is power", as commonly attributed to Sir Francis Bacon, is taught perhaps in the first class in China. My parents believe it, I believe it, and I will pass the words to my children in the future.

I feel grateful to Dr Nauman Aslam and Dr Hubert P. H. Shum who accepted me to the Erasmus Mundus gLINK international exchange project in 2015. It is this project that financially supported me to conduct my PhD study at Northumbria University between October 2016 and December 2018. During this time, Hubert is my principal supervisor and Nauman is my second supervisor. I thank them a lot for their valuable feedback on my research work, and I especially need to thank Hubert who has spent hours and hours of hard efforts on high-quality paper revisions before every submission. Every time, I was truly stunned by how energetic and critical Hubert was still even though he had finished several meetings before ours.

I must thank Adrien Papaioannou and Denis Papaioannou because without working for them in Tenokonda Ltd. since December 2018, I would not have been able to finance my study until now. I have enjoyed working for the company.

Crystal Wai-Ling Cheng, the love of my life, my fiancee, with whom I have spent the happiest period of time in my life since 21st October 2017. We are going to get married on the 4th February 2020, and I always feel blessed to have Crystal as the central part of my future life.

Special thanks are extended to my friends in the lab during the last three years, including Yijun Shen, Jingtian Zhang, Shoujiang Xu, Ying Huang, Jing Tian, Zemin Zuo, Daniel Organisciak, Kevin McCay, and Dimitrios Sakkos. We keep in touch!

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others. The work was done in collaboration with *Hubert P. H. Shum, Nauman Aslam, Xiaohui Liang, Frederick W. B. Li, and Antonio Mucherino*.

I declare that the word count of this thesis is 42280 words.

Name: Shanfeng Hu

Signature:

Date: 02 February 2020

Chapter 1

Introduction

2D and 3D geometric shapes are perhaps the most expressive digital channels through which real-world objects can be realistically represented, manipulated, animated, and displayed on computers. The applications of geometric shapes span a broad range of areas, such as computer graphics (Schneider and Eberly, 2002), computer-aided design (Farin, 2014), performance capture (Xia et al., 2017), 3D printing (Zhang et al., 2016), virtual reality (Zhao, 2009), and augmented reality (Billinghurst et al., 2015). Driven by the rapid advances in 3D scanning and sensing technologies (Zhang, 2012; Hitomi et al., 2015), shape analysis has become one of the central and most intensively studied research topics in the past, which aims to extract semantically meaningful knowledge (e.g. salient points, semantic part labels, correspondence) from geometric shapes for understanding and retrieval (Shilane et al., 2004; Bronstein et al., 2011; Zhao et al., 2014). Shape synthesis, as opposite to shape analysis, concerns with generating visually realistic geometric shapes (e.g. human faces and bodies) from external inputs for performance capture (Cao et al., 2015), animation (Loper et al., 2015), and image-based 3D reconstruction (Blanz et al., 1999). Recently, powerful deep learning methods have also been introduced to analyse and synthesise shapes (Wu et al., 2015; Qi, Su, Mo and Guibas, 2017*a*; Wu et al., 2017; Gkioxari et al., 2019). Therefore, addressing the two fundamental research topics not only enhances our theoretical understanding of how real-world shapes are organised and distributed, but also has significant impacts on diverse practical applications.

1.1 Motivation

In this thesis, we propose a unified metric framework to address three important yet diverse applications in shape analysis and synthesis: 2D facial shape beautification in images (Leyvand et al., 2008), salient points detection for 3D shapes (Lee et al., 2005), and non-rigid matching (i.e. correspondence finding) for 3D shapes (Van Kaick et al., 2011). The underlying theme of our solutions to the three problems is that we represent a shape as a square symmetric matrix that characterises the pairwise geometric relationships among all the points on the shape. The foremost motivation of our metric representations is that they are able to capture the global geometric features of shapes for holistic analysis and synthesis, which are more effective than traditional local feature representations that only consider individual points (Meyer et al., 2003) or surface regions (Belongie et al., 2001; Tombari et al., 2010). While recently there have been some metric representations proposed for geometry processing (Boscaini, Eynard, Kourounis and Bronstein, 2015; Corman et al., 2017; Corman and Ovsjanikov, 2019), they actually encode raw mesh connectivity information (e.g. adjacency, edge lengths, dihedral angles) and cannot be directly used for high-level semantic analysis and synthesis. Instead, our metric representations are derived from higher-level shape features and therefore enable direct semantic knowledge extraction and user-controlled synthesis.

The second motivation of our metric representations is that they are symmetric and therefore induce a set of mutually orthogonal, semantically informative eigenvectors for shape analysis and synthesis. Concretely, our representations formulate 1) facial shape synthesis as the projection onto the eigenvectors of an orthogonal projection metric, 2) saliency detection as computing the sparse principal eigenvector of a feature distance metric, and 3) non-rigid saliency matching as computing the Laplacian eigenvectors of a learned feature distance metric. As these eigenvectors can be optimally computed, our representations fundamentally admit global optimisation for the three tasks.

The last but not the least motivation of our metric representations is that they 1) allow us to incorporate flexible user control in facial shape beautification (e.g. preserving the shape of lips while enhancing the shape of jawline), 2) provide a theoretical understanding of how saliency can be well-defined on 3D shapes to accurately explain ground-truth eye fixations, and 3) enable saliency detection and non-rigid shape matching to help each other generalise better in a deep multi-task metric learning framework.

Recently, there are an increasing amount of research work on modelling human body shapes and poses for realistic human synthesis (Loper et al., 2015), which has extensive applications in computer gaming, performance capture, and virtual reality industries. Therefore, we are motivated to generalise shape synthesis to the joint modelling of human shapes and poses as an extension work of this thesis. Our key idea is learning the high-dimensional probability distributions of human shapes and poses in a much lower-dimensional auto-encoding space, where we use generative adversarial networks (GANs) to approximate the manifolds of real-scanned humans. The capacity of our learned human priors is best applied to virtual human synthesis in games (Zyda, 2005).

1.2 Problem Statement

The central question we want to ask in this thesis is: *can we find a global representation of 2D and 3D shapes for shape analysis and synthesis that are globally optimal in some mathematical sense*? To approach this question, we focus on the following three specific problems that have wide applications in computer graphics:

- **Mesh Saliency Detection**: Given an input 3D surface mesh, compute a continuous saliency map that highlights the semantically important points on the surface.
- **Unification of Saliency Detection and Non-rigid Shape Matching**: Given a pair of 3D surface meshes of the same object category (e.g. human body, four-leg animal), compute two respective saliency maps as well as a discrete correspondence map that pairs up semantically similar points on the two surfaces.
- **Facial Shape Beautification**: Given an input facial image, synthesise an output image with improved facial attractiveness by modifying the facial shape to a small extent. Users should be allowed to flexibly control the level of beautification for individual facial parts (e.g. jawline, lips, nose, and eyes).

As an extension of the work on shape synthesis, we are also interested in synthesising realistic-looking human shapes and poses at the same time. Therefore, we also consider the problem of human shape and pose modelling in this thesis. In particular, given a pair of human shape and pose datasets, the task is estimating the two corresponding probability distributions so that random sampling from them can synthesise realistic-looking human shapes and poses.

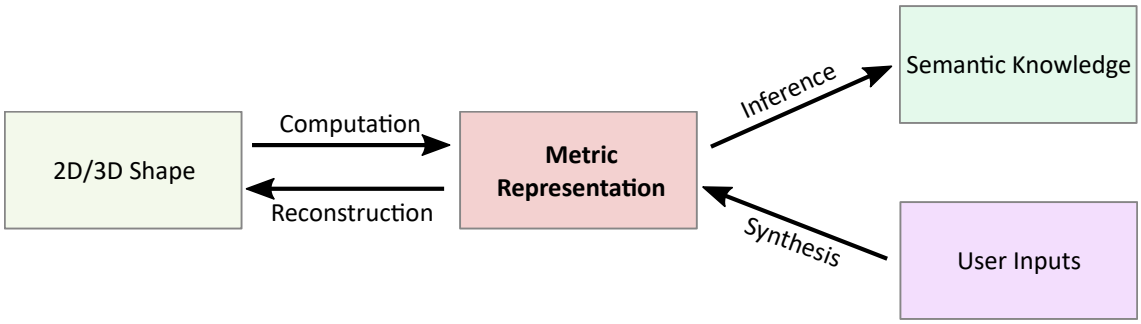


Figure 1.1: **The Overview of Our Unified Metric Framework**. For shape analysis, our framework computes a metric representation from input shapes and then derives semantic knowledge from the metric using optimisation. For shape synthesis, it synthesise a metric representation from external user inputs and then reconstructs target shapes using optimisation.

1.3 Our Methodology

As illustrated in Fig. 1.1, the central theme of our methodology is formulating shape analysis and synthesis tasks based on a metric representation. For shape analysis, the benefit of having a metric representation is that it captures the global geometric features of shapes, which allows us to infer semantic knowledge solely from the metric using global optimisation. For shape synthesis, the benefit is that a metric representation allows users to control the global geometric features of generated shapes while reconstructing output shapes using global optimisation. In the following, we substantiate this idea into the three problems we focus on in this thesis respectively. Meanwhile, we also describe our extension work on the joint modelling of human shapes and poses, which represents a continuum of the work on shape synthesis.

- **Metric-based Mesh Saliency Detection**: In this work, we propose an accurate and robust approach to salient region detection for 3D polygonal surface meshes. The salient regions of a mesh are those that geometrically stand out from their contexts and therefore are semantically important for geometry processing and shape analysis. However, a suitable definition of region contexts for saliency detection remains elusive in the field, and the previous methods fail to produce saliency maps that agree well with human annotations. We address these issues by computing saliency in a global manner and enforcing sparsity for more accurate saliency detection. Specifically, we represent the geometry of a mesh using a metric that globally encodes the shape distances between every pair of local regions. We then propose a sparsity-enforcing rarity optimisation problem, solving which allows us to obtain a

compact set of salient regions globally distinct from each other. We build a perceptually motivated 3D eye fixation dataset and use a large-scale Schelling saliency dataset for extensive benchmarking of saliency detection methods. The results show that our computed saliency maps are closer to the ground-truth. To showcase the usefulness of our saliency maps for geometry processing, we apply them to feature point localisation and achieve higher accuracy compared to established feature detectors.

- **Metric-based Unification of Saliency Detection and Non-rigid Shape Matching**: In this research, we propose a deep metric for unifying the representation of mesh saliency detection and non-rigid shape matching. While saliency detection and shape matching are two closely related and fundamental tasks in shape analysis, previous methods approach them separately and independently, failing to exploit their underlying relation that can mutually benefit each other. In view of the existing gap between saliency and matching, we propose to solve them together using a unified metric representation of surface meshes. We show that saliency and matching can be rigorously derived from our representation as the principal eigenvector and the smoothed Laplacian eigenvectors respectively. Learning the representation jointly allows matching to improve the deformation-invariance of saliency while allowing saliency to improve the feature localisation of matching. To parameterise the representation from a mesh, we also propose a deep recurrent neural network (RNN) for effectively integrating multi-scale shape features and a soft-thresholding operator for adaptively enhancing the sparsity of saliency. Results show that by jointly learning from a pair of saliency and matching datasets, matching improves the accuracy of the detected salient regions on meshes, which is especially obvious for small-scale saliency datasets, such as those having one to two meshes. At the same time, saliency improves the accuracy of shape matchings among meshes with reduced matching errors on surfaces.

- **Metric-based Facial Shape Beautification**: In this research, we propose a method to automatically beautify the shapes of non-frontal, non-neutral faces in photos, while allowing users to prescribe beautification weights for individual facial features. Previous methods focus on the beautification of mostly frontal and neutral faces, without incorporating user controllability in the beautification process. To address these restrictions, we propose the facial metric representation, which is affine-invariant, captures the pairwise geometric configura-

tion of facial landmarks, and allows for efficient face beautification with the user-specified weights of individual facial parts. We also propose an unsupervised beautification method in the operator space of faces, where an input face is iteratively pulled towards a local nearby density mode with improved attractiveness. Our method does not require human-annotated attractiveness scores for beautification. It also preserves the original poses and expressions for beautified faces. Results show that our method improves facial attractiveness for a range of poses and expressions, while retaining certain facial features according to user requirements.

- **From Shape to the Joint Modelling of Shape and Pose**. In this study, we propose to extend shape synthesis to the joint synthesis of shapes and poses for humans. The prior knowledge of real human body shapes and poses is fundamental in computer games and animation (e.g. performance capture). Linear subspaces such as the popular SMPL model have a limited capacity to represent the large geometric variations of human shapes and poses. What is worse is that random sampling from them often produces non-realistic humans because the distribution of real humans is more likely to concentrate on a non-linear manifold instead of the full subspace. Towards this problem, we propose to learn human shape and pose manifolds using a more powerful deep generator network, which is trained to produce samples that cannot be distinguished from real humans by a deep discriminator network. In contrast to previous work that learn both the generator and discriminator in the original geometry spaces, we learn them in the more representative latent spaces discovered by a shape and a pose auto-encoder network respectively. Random sampling from our priors produces higher-quality human shapes and poses. The capacity of our priors is best applied to applications such as virtual human synthesis in games.

1.4 Our Contributions

Our key contributions in this thesis are as follows:

- We propose a new metric representation of 2D/3D shapes for shape analysis and synthesis. Different from the traditional representations such as meshes, parameterisations, and Laplacians, our representation captures the global pairwise relationship among all points of

a shape, which leads to solutions from principled optimisations rather than hand-crafted yet suboptimal computation rules.

- To demonstrate the usefulness of our metric representation for shape analysis, we propose a sparse metric-based rarity optimisation method for saliency detection in Chapter 3. Our method is shown to produce more accurate saliency maps compared to the competing methods without relying on human annotations. Following this work, we then propose to unify saliency detection and non-rigid shape matching via a jointly learned metric representation in Chapter 4. The results show that matching improves the accuracy and deformation-invariance of saliency via the intra-category consistency of matching, while saliency improves the robustness of matching under non-isometric deformations via the sparsity of saliency.

- To demonstrate the usefulness of our metric representation for shape synthesis, we propose to represent the geometry of a face using the orthogonal projection metric onto the subspace of the facial landmarks for 2D facial shape beautification in Chapter 5. Our method is shown to improve the attractiveness of both frontal-neutral and non-frontal-non-neutral faces in the user studies. We also propose to learn the non-linear manifolds of real human shapes and poses in the low-dimensional auto-encoding spaces rather than in the original high-dimensional geometry spaces in Chapter 6. We demonstrate the capacity of our learned priors by generating high-quality human shapes and poses via random sampling.

- We make our datasets and source codes from this thesis publicly available for reproduction and validation. Our 3D eye fixation dataset and saliency detection source codes can be downloaded from here* and here† respectively. Our source codes for unifiying saliency detection and shape matching can be downloaded from this link‡. Our face beautification source codes can be downloaded from this link§. Our human shape and pose modelling codes can be downloaded from this link¶.

*https://drive.google.com/open?id=1k88SJOGEGSUKneDQHvBHp5TE2CROA4DK
†https://drive.google.com/open?id=14gzatUYMeRGEb0F_3GOihqD3Y0BojZFF
‡https://drive.google.com/drive/folders/10Vu3ujF-5gPm8h_E35VhZR45WCjht18B
§https://drive.google.com/open?id=1NonS5WQedtxejTDh-m_Ym3MZSk21H54p
¶https://drive.google.com/open?id=1y-aPe8FGzetxnY3FpSESci3U59KgQSUJ

1.5 Thesis Structure

The structure of this thesis is as follows. Chapter 2 surveys the related work in the field, including shape representations in Section 2.1, mesh saliency detection in Section 2.2.1, non-rigid shape matching in Section 2.2.2, facial shape beautification in Section 2.3.1, as well as human shape and pose synthesis in Section 2.3.2.

Chapter 3 describes our proposed metric-based approach to mesh saliency detection. The chapter starts with introducing the problem background, motivations, and our contributions in Section 3.1. It then presents an overview of our saliency detection approach in Section 3.2.1. Following the order of the four key components of our approach, it presents Mesh Sampling in Section 3.2.2, Shape Descriptor Construction in Section 3.2.3, Sparse Metric-based Rarity Optimisation in Section 3.2.4, and finally Vertex Saliency Interpolation in Section 3.2.5. After presenting our approach, the chapter then proceeds to how we constructed our 3D eye fixation dataset for methods evaluation in Section 3.3. In Section 3.4, the chapter first shows some saliency maps computed by our approach in Section 3.4.1, and then presents the visual and quantitative comparisons of our approach with others in section 3.4.2 and 3.4.3 respectively. After that, the chapter presents the robustness comparison of our approach with others in Section 3.4.4, and also compares our approach with established methods for feature points localisation on 3D surface meshes in Section 3.4.5. The breakdown of the run-time of our approach is in Section 3.4.6. Finally, the chapter is concluded with the key findings in Section 3.5.

Chapter 4 presents our metric approach for unifying mesh saliency detection and non-rigid shape matching. In Section 4.1, the chapter introduces the two problems considered, motivates why they should be jointly addressed for mutual benefits, and briefly describes our metric contributions for unifying them. In Section 4.2, the chapter elaborates our unified metric representation, with some mathematical notations defined in Section 4.2.1 as well as the metric-based inference methods for saliency detection and shape matching presented in Section 4.2.2 and 4.2.3 respectively. In Section 4.3, the chapter presents our proposed deep learning architecture for metric learning. Firstly, it describes the proposed deep RNN architecture of our approach for multi-scale metric learning in Section 4.3.1. Secondly, it describes the soft-thresholding operator of our approach for metric sparsification in Section 4.3.2. Lastly, it describes and analyses the unified loss function of our approach for metric learning in Section 4.3.3. In Section 4.4, the chapter begins by giving the im-

plementation details of our approach in Section 4.4.1. It then presents the quantitative evaluation results of our deep learning architecture in Section 4.4.2, and demonstrates how saliency detection and shape matching mutually improves each other in Section 4.4.3. After that, the chapter presents the visual and quantitative comparisons of our approach with others for the two tasks in Section 4.4.4 and 4.4.5 respectively. Finally, in Section 4.5, the key findings from this study are discussed.

Chapter 5 presents our metric approach for the task of facial shape beautification in images. Section 5.1 introduces the problem background and motivates our metric representation for facial shape synthesis. Section 5.2 elaborates our approach, which begins with facial landmarks detection in Section 5.2.1 and then proceeds to our proposed facial metric representation in Section 5.2.2. After that, Section 5.2.3 describes how an input facial metric can be beautified using mean-shift clustering and Section 5.2.4 shows how the beautified version of the input facial landmarks can be reconstructed from the beautified metric. In Section 5.2.5, the method of generating beautified facial images is presented. Section 5.3 presents the experimental results, which include full-face beautification results in Section 5.3.1, quantitative user study results in Section 5.3.2, user-controlled beautification results in Section 5.3.3, and run-time information in Section 5.3.4. Finally, the chapter is concluded in Section 5.4.

Chapter 6 presents our extension work on the joint modelling of realistic human shapes and poses. In this chapter, Section 6.1 introduces the problem background and our key contributions. Section 6.2 describes our proposed system, where we first elaborates shape and pose embedding in Section 6.2.1, then presents our approach of generative modelling in the auto-encoding space in Section 6.2.2, and finally list the implementation details in Section 6.2.3. After that, Section 6.3 presents the experimental results and Section 6.4 concludes the chapter.

Chapter 7 summarises the key contributions of this thesis and discusses the future work for saliency detection in 7.1, non-rigid shape matching in 7.2, and facial shape beautification in 7.3.

Chapter 2

Literature Review

In this chapter, we review the existing work on shape representations in Section 2.1, on shape analysis in Section 2.2, and on shape synthesis in Section 2.3. For shape analysis, we review mesh saliency detection and non-rigid shape matching in Section 2.2.1 and 2.2.2 respectively. For shape synthesis, we review facial shape beautification and human shape and pose modelling in Section 2.3.1 and 2.3.2 respectively. We discuss the existing methods and highlight the contributions of our proposed ones for shape analysis and synthesis.

2.1 Shape Representations

The representation of geometric shapes is perhaps the most fundamental concept in computer graphics, geometry processing, and shape analysis. There have been different levels of shape representations proposed in the past, with their respective most suitable applications. Here, we broadly classify them into the following categories and highlight the difference of our proposed metric representation with them.

Polygonal Meshes. Perhaps the most popular shape representation in computer graphics is the polygonal mesh, which is the standard format for computer graphics rendering system such as OpenGL and DirectX. A mesh consists of a set of vertices and a set of edges that connect the adjacent vertices (Botsch et al., 2010), sometimes with the subdivision structure (DeRose et al., 1998) or the progressive structure (Hoppe, 1996) to capture varying levels of geometric details. Polygonal meshes are most suitable for graphics rendering and differential properties computation

(Meyer et al., 2002) because of their low-level structure. However, they are not convenient for high-level shape analysis because it remains very difficult to infer semantic knowledge directly from collections of vertices and edges. Differently, our metric representation is free of the low-level topology constraints and directly captures the pairwise relationship among all points of a shape, which as a result enables more efficient shape analysis and synthesis.

Parameterisations. Another very popular representation of shapes are their parameterisations, which can be on the rectangular domain (Gu et al., 2002), the spherical domain (Sheffer et al., 2004), the cube domain (Tarini et al., 2004), or arbitrary domains (Kraevoy and Sheffer, 2004). These parameterisations are essentially equivalent to their polygonal mesh counterparts and most suitable for texturing and storing. However, they still encode low-level geometry details and remain difficult for shape analysis and synthesis. In contrast, our metric representation is free of the underlying domain selection and therefore focuses more on the domain-invariant semantic knowledge of shapes.

Laplacians. The Laplacian of a mesh is a square symmetric matrix that captures the geometric relationship among locally adjacent vertices. Applying the Laplacian on the vertex coordinates produce the so called differential coordinates that represent the displacement from each vertex to the mean position of its one-ring neighbours (Belkin et al., 2008). As the differential coordinates encode the local surface details, the Laplacian is widely used for mesh editing (Sorkine et al., 2004) and mesh optimisation (Nealen et al., 2006). Also, the eigenvectors of the Laplacian capture the geometry details of varying frequencies and thus are extensively used for surface filtering (Vallet and Lévy, 2008) and spectral processing (Zhang et al., 2010). Although our metric representation is also of the square matrix form, it is different from the Laplacian in that it captures the global pairwise relationship among all points of a shape, rather than the relationship among locally adjacent points. Recently, there has been work on recovering shapes from given Laplacians (Boscaini, Eynard, Kourounis and Bronstein, 2015). However, due to the low-level nature of the Laplacian, the method cannot incorporate user inputs in the reconstruction process and requires iterative optimisation. In contrast, our metric representation admits high-level user designs and permits much more efficient shape reconstruction.

Semantic Representations. More recently, there have been machine learning and deep learning methods proposed for inferring high-level abstract feature representations from low-level geomet-

ric data for semantic shape analysis. The features can be learned from raw point clouds (Qi, Su, Mo and Guibas, 2017*a*; Qi, Yi, Su and Guibas, 2017), geometry images (Sinha et al., 2016), polygonal meshes (Masci et al., 2015), or Laplacians (Litman and Bronstein, 2014; Boscaini, Masci, Rodolà, Bronstein and Cremers, 2016; Boscaini, Masci, Melzi, Bronstein, Castellani and Vandergheynst, 2015). They have been used for a number of tasks such as classification and segmentation but remain limited for the task of mesh saliency detection we focus on in this thesis. This is because they mostly formulate shape analysis as per-point regression or classification, without correlating all points together for a holistic analysis. In contrast, our metric representation derives the saliency of all points as the principal eigenvector of a metric and therefore naturally allows for a holistic and more accurate analysis. Also, the previous feature representations are mostly used for analysis rather than synthesis, while our metric representation enables shape reconstruction from a given metric effortlessly.

2.2 Shape Analysis

2.2.1 Mesh Saliency Detection

Visual Attention Modelling

The theoretical foundation of visual attention can be traced back to (Treisman and Gelade, 1980), where Treisman and Gelade proposed "Feature-Integration Theory" which suggests what and how visual features are combined to direct human visual attention. A feed-forward computational method to incorporate these features was developed by (Koch and Ullman, 1987), indicating that salient image locations are visually distinct from their surroundings. A centre-surround operator on low-level image features was implemented by (Itti et al., 1998) for saliency detection. After that, a large body of visual saliency methods have been proposed in computer vision (Harel et al., 2006; Hou and Zhang, 2007; Ji et al., 2019; Borji and Itti, 2013). Our proposed method is also for visual attention modelling but works on 3D surface meshes instead of 2D images.

Saliency Detection for 3D Scenes

To accelerate realistic rendering, (Yee et al., 2001) used the method of (Itti et al., 1998) to detect salient regions of coarsely rendered scenes and focused rendering resources on these important

regions. Similar to (Yee et al., 2001), (Longhurst et al., 2006) controlled per-pixel ray sampling density based on detected salient regions. Afterwards, (Sundstedt et al., 2007) extended the idea to participating media rendering and achieved realistic results with low computational costs. (Mantiuk et al., 2003) made an attempt to compress animated scenes with the guidance of image saliency. By only using salient regions of rendered images, these methods have no access to any depth information of 3D scenes, which plays an important role in human visual attention (Howard, 2002). In contrast, our method analyses 3D geometry directly and therefore can detect structure-related saliency information.

Saliency Detection for Surface Meshes

We classify existing mesh saliency detection methods into the following five categories:

Local Contrast. Inspired by (Itti et al., 1998), (Lee et al., 2005) introduced the concept of mesh saliency using a centre-surround operator on Gaussian-weighted mean curvatures. (Gal and Cohen-Or, 2006) defined the saliency of a region based on its relative size, curvatures, and curvature changes. They detected and segmented salient regions for partial shape matching. (Feixas et al., 2009) proposed an information-theoretical framework for viewpoint selection and mesh saliency computation. (Zhao et al., 2016) computed saliency from local normal information for subsequent refinement. These models compute saliency as local contrast of surface properties, generally by comparing local regions to their neighbours. (Jeong and Sim, 2017) used normal information to compute both view-independent and view-dependent saliency. However, local contrast methods tend to wrongly identify bumpy and noisy regions as salient. Our method addresses this issue by extending the region comparison from a local to a global context.

Spectral Irregularity. Based on the work of (Hou and Zhang, 2007), (Song et al., 2014) assumed that saliency is hidden in the irregularity of the log-Laplacian spectrum of a mesh. They extracted such irregularity and transformed it back to the spatial domain to compute saliency. Mesh simplification (Garland and Heckbert, 1997) was used to tackle the costly eigendecomposition of a large Laplacian matrix. Due to the spatial unawareness of spectral basis, this method has difficulties in capturing some individual local salient regions. Our method, instead, works in the spatial domain and can capture salient regions from small to large scales.

Shape Discrimination. Analogous to feature selection in classification problems, (Shilane and

Funkhouser, 2007) detected salient surface regions for distinguishing shapes of different object categories. The detection results depend not only on a semantically categorised shape database but also on the object categories of input meshes. In many applications, however, these semantic annotations remain scarce and their collection is labour-intensive. Our method does not require any semantic data for saliency detection, thereby allowing the use for geometry processing applications that do not have semantic annotations.

Learning-based Detection. Through a large-scale online user study, (Chen et al., 2012) obtained massive amounts of salient surface points for a library of meshes. They used the obtained data to train regression models for saliency prediction. A variety of low-, middle-, and high-level cues, such as curvatures, geodesics, segmentation, and symmetries, were incorporated. The trained models performed well on the training set but showed limited generalisation abilities for novel meshes out of training datasets. Our method does not require training before use and shows good generalisation to diverse object categories.

Global Rarity. (Leifman et al., 2012) detected more globally rare surface regions for viewpoint selection. (Wu et al., 2013) combined local contrast with global rarity to compute mesh saliency. (Pingping et al., 2015) detected salient regions by identifying non-salient backgrounds via manifold ranking. These methods aim to suppress repeated patterns across a mesh surface by extending region comparison from a local to a wider context. However, they require mesh segmentation and geodesic distance computation which are not robust to topological flaws such as holes and non-manifold structures of meshes. Our method belongs to this category and is based on a global metric representation, which is robust to underlying potentially poor mesh tessellations.

Mesh Saliency Evaluation

(Howlett et al., 2005) computed saliency maps from human eye fixations to guide mesh simplification. They demonstrated that preserving salient details could improve the fidelity of simplified meshes. Using eye-tracking experiments, (Kim et al., 2010) validated that saliency (Lee et al., 2005) was better compared to mean curvature for eye fixation prediction. We extend the eye-tracking experiment of (Kim et al., 2010) from 2D to 3D and build a 3D eye fixation dataset suitable for public saliency detection benchmarking.

Mesh Saliency Applications

Numerous graphics applications have benefited from mesh saliency. Kim and Varshney used saliency to edit surface (Kim and Varshney, 2008) and volume (Kim and Varshney, 2006) regions for highlighted visualization. (Liu et al., 2007) and (Miao and Feng, 2010) employed saliency to detect feature points and extremum lines for mesh segmentation and depiction. Recently, (Gu et al., 2014) combined saliency with Poisson sampling for adaptive depth image compression. Other applications include mesh simplification (Lee et al., 2005; Shilane and Funkhouser, 2007; Wu et al., 2013; Song et al., 2014), viewpoint selection (Lee et al., 2005; Shilane and Funkhouser, 2007; Leifman et al., 2012; Secord et al., 2011), shape matching (Gal and Cohen-Or, 2006; Shilane and Funkhouser, 2007), mesh sampling (Wu et al., 2013), surface reconstruction(Song et al., 2014), and crowd modelling (McDonnell et al., 2009). We apply our method to the task of feature point detection which is a fundamental building block in many geometry processing applications.

2.2.2 Non-rigid Shape Matching

Model-based Shape Matching

Non-rigid shape matching finds semantically meaningful surface correspondences across meshes irrespective of the deformations among them (Van Kaick et al., 2011). Traditional methods mainly assumed the deformation to be isometric (Ovsjanikov et al., 2012) or conformal (Kim et al., 2011) and then searched for matchings within the prescribed deformation space. The former assumes that the deformation between a pair of meshes preserve the curve lengths on the surfaces, while the latter assumes that only the angles on the surfaces are maintained. Due to the isometry-invariant property, the surface Laplacian (Rustamov, 2007) has been widely used in the spectral embedding (Sun et al., 2009), functional mapping (Ovsjanikov et al., 2012), and quadratic matching (Maron et al., 2016) formulations of shape matching. Both isometric and conformal deformations, however, are overly restricted and can bias shape matching towards unfavourable solutions.

Learning-based Shape Matching

Recent methods learned deformation-invariant shape embeddings for correspondence search (Corman et al., 2014; Litman and Bronstein, 2014; Boscaini, Masci, Rodolà, Bronstein and Cremers,

2016; Cosmo et al., 2016; Wei et al., 2016) or directly learned point label classifiers for correspondence prediction using random forests (Rodolà et al., 2014), convolutional neural networks (CNNs) (Boscaini, Masci, Melzi, Bronstein, Castellani and Vandergheynst, 2015; Boscaini, Masci, Rodolà and Bronstein, 2016), and multi-layer perceptrons (MLPs) (Litany et al., 2017). Both streams of methods learned for individual points without considering their saliency information. Also, the first stream of methods learned embeddings on a per-point basis and lacked orthogonality and smoothness guaranties that hold for the Laplacian embeddings (Rustamov, 2007). Our method instead guarantees that the learned embeddings are orthogonal and smooth. More importantly, it exploits saliency to ensure that the resulting embeddings are localised on semantically important surface regions. This is particularly valuable in improving both the model-based and learning-based methods for matching non-isometric pairs of shapes.

2.3 Shape Synthesis

2.3.1 Facial Shape Beautification

Facial Modelling and Editing

One approach to face beautification in natural photos is reconstructing 3D faces from 2D images and applying the 3D face rectification method of (Liao et al., 2012). Despite the development of statistical shape models (Blanz and Vetter, 1999; Tena et al., 2011; Maleš et al., 2019) and example-based models (Kemelmacher-Shlizerman et al., 2011; Hassner, 2013), 3D face reconstruction with a wide range of poses and expressions remains a challenging ill-posed problem. While the work of (Yang et al., 2011) can be used for facial component transfer, it does not address the problem of face beautification. Therefore, following (Leyvand et al., 2008; Chen et al., 2014), we focus on 2D face beautification in this work.

Textural, expressive, and photometric traits also have important influence on the perception of facial attractiveness. As a result, a large body of research have been devoted to the editing of these traits in 2D images, such as face makeup (Guo and Sim, 2009; Scherbaum et al., 2011; Zhang et al., 2019), expression editing (Yang et al., 2011, 2012), pimples removal (Brand and Pletscher, 2008), photometry correction (Joshi et al., 2010), and hair decoration (Pasupa et al., 2019). Our work complements these methods in that we focus on editing the geometric trait of faces in 2D

images.

Facial Attractiveness Analysis

Computer techniques have been used for facial attractiveness analysis over two decades. We refer the readers to (Laurentini and Bottino, 2014) for an excellent review of the field. (Grammer and Thornhill, 1994) composed different facial images to verify the effect of symmetry and averageness on the perception of facial attractiveness. Later, (Zhang et al., 2011) validated the effect of averageness using geometrically transformed faces. (Schmid et al., 2008) took a rule-based approach to examine that human-annotated attractiveness scores were consistent with that predicted by neoclassical canons, symmetries, and golden ratios. (Eisenthal et al., 2006) represented the first to use machine learning for facial attractiveness prediction. They employed human subjects to rate a library of faces and trained attractiveness regression models using the appearance and geometry features of faces. Following this, a number of learning-based facial attractiveness prediction methods have been proposed (Zhang et al., 2017; Gan et al., 2014; Chen et al., 2014). As it is ambiguous to rate the attractiveness of non-frontal, non-neutral faces, previous work mostly focuses on the analysis of frontal portraits. Therefore, we propose an unsupervised beautification method for non-frontal, non-neutral faces.

Facial Attractiveness Enhancement

Compared with facial attractiveness analysis, the enhancement problem has received relatively less research attention. (Liao et al., 2012) optimised the shape of a 3D face model by deforming it towards the beauty cannons summarised by (Schmid et al., 2008). The method works well for neutral faces but has inherent difficulties of generalising to non-neutral faces. (Leyvand et al., 2008) was the first to approach attractiveness enhancement using machine learning. Their results validated the feasibility of a data-driven approach to face beautification. Recently, (Chen et al., 2014) searched for a convex combination of attractive faces while maximising its resemblance to the original face. However, both methods require human annotations of attractiveness scores for training and cannot work for non-frontal, non-neutral faces, which are actually the main subjects of real-world photos. Deep learning has also been applied to facial attractiveness enhancement (Li et al., 2015). However, the resulting system does not allow for the controllability of beautification due to the black-box nature of deep learning techniques. In contrast from the methods of

(Leyvand et al., 2008; Chen et al., 2014; Li et al., 2015), our proposed one works for faces with non-frontal poses and non-neutral expressions in images. Benefiting from the linearity of our metric representation, our method easily allows users to preserve certain facial parts while beautifying the remaining.

2.3.2 Human Shape and Pose Modelling

Here, we briefly discuss existing work on modelling the prior distributions of human body shapes and poses. Because human shapes and poses have geometric regularities, they are commonly assumed to be lying on the underlying low-dimensional manifolds, which are embedded in the original geometry ambient spaces. Linear subspace methods assume the shape and pose manifolds to be linearly embedded in the geometry spaces (Blanz et al., 1999; Loper et al., 2015), which can be computed using the linear PCA dimension reduction method. The drawback of linear methods is that they mostly use Gaussian as the generating distributions, which are globally supported on the full subspace. Sampling in the full subspace may produce non-realistic results (Kanazawa et al., 2018). More recent work makes a more realistic assumption of human shape and pose manifolds, that they are non-linearly embedded in the geometry spaces. The variational auto-encoding (VAE) framework of (Kingma and Welling, 2013) has been exploited to learn human shape and pose manifolds in (Tan et al., 2018) and (Habibie et al., 2017) respectively. There are also methods using GANs (Goodfellow et al., 2014) to model human shape and pose manifolds (Gokaslan et al., 2018; Kanazawa et al., 2018; Chen et al., 2017). We prefer GANs over VAE for non-linear distribution modelling in this work because the former have been extensively validated to be capable of producing sharp samples (e.g. images and shapes), while the latter tend to produce over-smoothed results. The main contribution of this work is our finding that learning the shape and pose manifolds in the auto-encoding spaces with GANs produces high-quality samples.

Chapter 3

Metric-based Mesh Saliency Detection

In this chapter, we propose a metric representation approach for the shape analysis problem of mesh saliency detection. As shown in Fig. 3.1, the key feature of our approach is that we derive a saliency map not from the input mesh but from a feature distance metric computed from it. This allows us to gain a theoretical understanding of how saliency can be well-defined on 3D shapes, which we elaborate in the following sections.

3.1 Introduction

The human visual system has a remarkable ability to quickly and effortlessly identify a small number of interesting objects in a visual field. This ability appears mainly stimulus-driven and is commonly referred to as *visual attention*, which helps suppress the vast amount of visual inputs that are not essential to subsequent cognitive processing tasks (Borji and Itti, 2013).

In computer vision, numerous computational methods have been proposed to mimic the visual attention mechanism for efficient image understanding (Itti et al., 1998; Harel et al., 2006; Hou and Zhang, 2007; Ji et al., 2019). Inspired by the idea of saliency-guided image processing, (Lee et al., 2005) introduced the concept of *mesh saliency* to computer graphics, highlighting its advantages over traditional geometric quantities (e.g. curvatures) for assessing the perceptual importance of mesh regions. By prioritising the processing of mesh regions according to their saliency values,

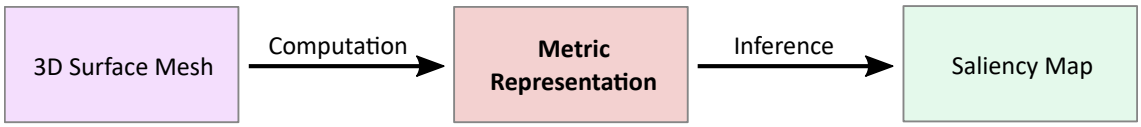


Figure 3.1: **The Overview of Our Unified Metric Framework for Mesh Saliency Detection**. Given an input 3D surface mesh, our framework first computes a metric representation and then derives a saliency map from the metric using global optimisation.

the perceptual quality of processed meshes can be largely retained and the processing time can be effectively reduced. Such examples include shape simplification (Lee et al., 2005; Shilane and Funkhouser, 2007; Wu et al., 2013; Song et al., 2014), shape matching (Gal and Cohen-Or, 2006; Shilane and Funkhouser, 2007), realistic rendering (Kim and Varshney, 2006, 2008; Miao and Feng, 2010; Sundstedt et al., 2007), shape segmentation (Liu et al., 2007; Song et al., 2014), shape reconstruction (Song et al., 2014), and crowd modelling (McDonnell et al., 2009).

Despite the vast use of mesh saliency in geometry processing and computer graphics, the definition of what constitutes saliency remains elusive in the field. The fundamental challenge is that the attention mechanism of human vision is far from being fully understood (Borji and Itti, 2013). Regarding this, many efforts have been devoted to hand-crafting some computational methods that take a 3D polygonal mesh as input and produce a saliency map as output. For example, the local contrast methods of (Lee et al., 2005; Gal and Cohen-Or, 2006; Feixas et al., 2009) compute the saliency of a local region as the difference of its differential properties from its surroundings. The global rarity methods of (Leifman et al., 2012; Wu et al., 2013; Pingping et al., 2015; Song et al., 2014) compute the saliency of regions in wider surroundings and are able to highlight more distinct shape features. However, the former mainly respond to local geometric variations and suffer from surface noises and bumps, while the latter are sensitive to topological flaws due to the use of mesh connectivity for saliency computation.

There are also methods using high-level semantic annotations for saliency computation. One example is the method of (Shilane and Funkhouser, 2007) that detects salient regions effective at distinguishing shapes of different object categories. The other example is the tree-regression-based method of (Chen et al., 2012), which learns to predict saliency from low- and high-level geometric properties such as curvatures and symmetries. However, both methods require category-specific human annotations for saliency computation and therefore cannot generalise to more object categories without annotations.

In view of the above challenges, we propose a new saliency detection method that does not require human annotations, generates accurate saliency maps much closer to ground-truth, and is robust to mesh noises, simplification and holes. Our method is mathematically derived from two fundamental principles of saliency: *rarity* and *sparsity*. The rarity principle regards those regions distinct from others to be salient, and the sparsity principle ensures that only a small number of truly distinct regions can pop out from the saliency computation process. Without enforcing sparsity while optimising rarity, the computed saliency maps would become overly uniform and very few regions can stand out and be correctly recognised as salient.

Specifically, we propose a sparse metric-based rarity optimisation problem for saliency computation. The optimisation variable of the problem is the optimal saliency map to be solved for, and the optimisation objective is the continuous rarity of a metric encoding the shape distance between every pair of local regions. We incorporate the sparsity principle of saliency by constraining the L0-norm of any feasible saliency map solutions. As a result, our saliency detection method amounts to solving a sparse eigenvalue problem (Yuan and Zhang, 2013), with the optimal saliency map being the sparse eigenvector of the metric of a mesh. By averaging multi-scale metrics into a scale-free metric, our method is able to discover a compact set of multi-scale salient regions from raw geometric features. By avoiding the use of mesh connectivity in metric representation, our method is robust to mesh flaws such as simplifications, noises, and holes.

To evaluate the performance of saliency detection methods, we build a perceptually motivated 3D eye fixation dataset from 50 graphics meshes and 8 human subjects through a 3D eye-tracking experiment. We also implement the highly cited saliency detection methods of (Lee et al., 2005; Leifman et al., 2012; Pingping et al., 2015), whose original source codes are not publicly available for large-scale quantitative benchmarking. We perform extensive evaluations on competing methods, such as (Lee et al., 2005; Leifman et al., 2012; Song et al., 2014; Pingping et al., 2015), using our eye fixation dataset and the Schelling saliency dataset of (Chen et al., 2012). The results show that our computed saliency maps are closer to the ground-truth than that generated by the competing methods of (Lee et al., 2005; Leifman et al., 2012; Song et al., 2014; Pingping et al., 2015). To showcase the usefulness of our computed saliency maps, we apply them to feature point localisation (Dutagaci et al., 2012) and compare to the established feature detectors of (Lee et al., 2005; Novatnack and Nishino, 2007; Castellani et al., 2008; Sun et al., 2009). The results show

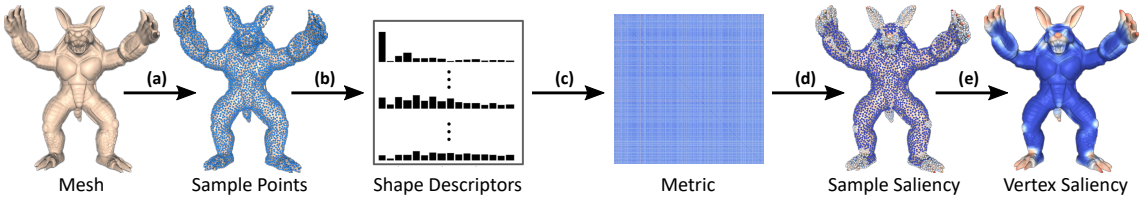


Figure 3.2: **The Overview of Our Metric-based Mesh saliency Detection Method**. The steps from (a)-(e) are mesh sampling, shape descriptor construction, metric computation, saliency optimisation, and vertex saliency interpolation respectively. We use red and blue colours to indicate high and low saliency values respectively.

that our saliency-guided feature detector outperforms others in terms of feature point localisation accuracy.

We propose three contributions in this study:

- We propose a sparse metric-based rarity optimisation method for saliency detection. Our method is shown to be able to produce accurate saliency maps without relying on human annotations while being robust to mesh simplifications, noises, and holes.
- We build a perceptually motivated 3D eye fixation dataset for saliency detection benchmarking. The dataset extends the eye fixation experiment of (Kim et al., 2010) from 2D to 3D, and complements the Schelling saliency dataset of (Chen et al., 2012) that is not constructed from real captured human eye movements. Our publicly available dataset can be downloaded from this link*.
- We perform a comprehensive evaluation of our approach and the distinctive previous methods of (Lee et al., 2005; Leifman et al., 2012; Pingping et al., 2015) on our eye fixation dataset and the Schelling saliency dataset of (Chen et al., 2012). The results show that our computed saliency maps are closer to the ground-truth annotations compared to that of (Lee et al., 2005; Leifman et al., 2012; Song et al., 2014; Pingping et al., 2015). Our open-source codes can be downloaded from this link†.

3.2 Our Metric Approach to Mesh Saliency Detection

3.2.1 Overview

We illustrate the computation steps of our proposed saliency detection method in Fig. 3.2. Given a 3D polygonal surface mesh as input, we first sample a set of random points on the surface (a). For each sample point, we construct a shape descriptor that characterises its local shape information (b). We then compute a matrix of squared Euclidean distances among all sample points using their shape descriptors (c). From this metric representation, we derive the optimal saliency map by solving a sparse metric-based rarity optimisation problem (d). Finally, we map the computed saliency from the sampled points back to the underlying mesh vertices via Gaussian filtering (e).

3.2.2 Mesh Sampling

To achieve the translation and uniform scaling invariance of saliency detection, we normalise an input mesh by locating its centroid at the origin and uniformly scaling its radius (i.e. the half diagonal length of its tight bounding box) to 1. As a surface mesh is sometimes either under- or over-tessellated, we randomly sample a set of points on the surface of the normalised mesh (Shilane and Funkhouser, 2007), so that the quality of computed saliency maps is maintained while the computational cost remains invariant to the original size of the mesh. We sequentially sample a triangle of the mesh with the probability proportional to the area of the triangle, and then generate a sample point on the triangle using randomly generated barycentric coordinates. We denote the sample point set as $\mathcal{P} = \{\boldsymbol{p}^i\}_{i=1}^N$ and empirically find that $N = 5000$ points are sufficient to cover the whole surface. We use this value in all of our experiments. We note that the fixed-size sampling method may have difficulty capturing the fine-scale geometric details of highly complex meshes or scenes. In the future work, we plan to adjust the sample size based on the ratio of the surface area of a mesh to the volume of its tight bounding box, so that sample points can be more efficiently allocated to sufficiently represent surface details.

As observed in (Lee et al., 2005; Shilane and Funkhouser, 2007), salient regions can range from small surface details to large surface parts. To accommodate the multi-scale nature of saliency,

*https://drive.google.com/open?id=1k88SJOGEGSUKneDQHvBHp5TE2CROA4DK
†https://drive.google.com/open?id=14gzatUYMeRGEb0F_3GOihqD3Y0BojZFF

we define a succession of increasingly larger regions for each sample point on the surface, $\mathcal{S} = \{0.02, 0.04, 0.06, 0.08, 0.1\}$. We denote a region $\boldsymbol{p}^{i,\tau}$ as a spherical volume of the radius $\tau \in \mathcal{S}$, with the volume centered at the sample point $\boldsymbol{p}^i \in \mathcal{P}$. We use this region representation because it is independent of the underlying potentially poor mesh tessellation (i.e. irregular meshes, non-manifold edges, and disconnected components) and easily supports multi-scale saliency computation by iterating through each scale in $\mathcal{S}$. We find that $\mathcal{S}$ works well for capturing both small- and large-scale salient features in practice.

3.2.3 Shape Descriptor Construction

For each region $\boldsymbol{p}^{i,\tau}$ defined in the above, we compute a feature vector $\boldsymbol{f}^{i,\tau}$ to characterise its local shape information. We choose the harmonic shape descriptor (Kazhdan et al., 2003) for three reasons. First, it is rotation-invariant and therefore does not require orienting a mesh before saliency computation. Second, it has a theoretically guaranteed minimal information loss, which is not met by other shape descriptors (Kortgen et al., 2003; Salti et al., 2014). Third, it has a spherical construction and naturally supports multi-scale saliency computation by adjusting the radius of the sphere.

To compute the descriptor, we convert a mesh into a Gaussian distance field of resolution $256 \times 256 \times 256$ and partition each region (i.e. a spherical volume) into 8 equally-spaced concentric shells (Shilane and Funkhouser, 2007). We sample the Gaussian distance field on these shells and compute the amplitudes of the first 8 spherical harmonic frequency bands for each shell (Kazhdan et al., 2003). Therefore, the shape descriptor length of each region is $8 \times 8 = 64$. We find that this feature granularity is sufficient to discriminate regions for effective saliency computation.

3.2.4 Sparse Metric-based Rarity Optimisation

Traditionally, the methods of (Lee et al., 2005; Gal and Cohen-Or, 2006; Feixas et al., 2009; Leifman et al., 2012; Wu et al., 2013; Pingping et al., 2015; Shilane and Funkhouser, 2007; Zhao et al., 2016; Jeong and Sim, 2017) compute saliency from some hand-crafted rules, lacking a principled goal of optimisation. In contrast, we derive saliency from optimising the rarity of a global metric representation while enforcing the sparsity of saliency. The rarity principle regards those regions that have the maximum distinction from others as salient, while the sparsity principle

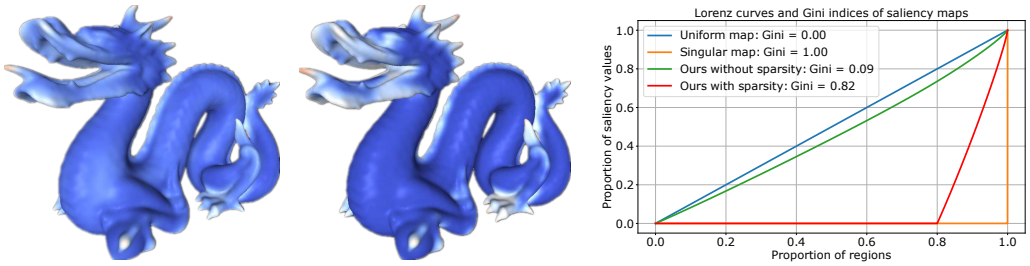


Figure 3.3: **Saliency Detection With and Without Sparsity**. *Left*: our computed saliency map without sparsity. *Middle*: our computed saliency map with sparsity. *Right*: Lorenz curves and Gini indices of saliency maps. Our map without sparsity is both visually and quantitatively similar to the uniform one, while our map with sparsity is quantitatively sparser and visually more concentrated on distinct regions.

ensures that only a compact set of truly distinct regions can stand out. To this end, we propose the following optimisation problem:

$$\arg\max R(\boldsymbol{\varphi}(\mathcal{P})) = \boldsymbol{\varphi}(\mathcal{P})^T \boldsymbol{M}(\mathcal{P})\boldsymbol{\varphi}(\mathcal{P}), \text{ s.t. } \boldsymbol{\varphi}(\mathcal{P}) \geq 0\,, \|\boldsymbol{\varphi}(\mathcal{P})\| = 1 \text{ and } \|\boldsymbol{\varphi}(\mathcal{P})\|_0 \leq \mu N \tag{3.1}$$

where $\boldsymbol{\varphi}(\mathcal{P}) \in \mathbb{R}^N$ is the saliency map of the sample points $\mathcal{P}$ to be solved for ($\boldsymbol{\varphi}_i(\mathcal{P})$ is the saliency value of point $\boldsymbol{p}^i$), and $\boldsymbol{M}(\mathcal{P}) \in \mathbb{R}^{N\times N}$ is the metric representation that encodes the pairwise shape contrasts among all sample points. Additionally, the first constraint ensures the solution saliency map to be element-wise non-negative, and the second constraint $\|\boldsymbol{\varphi}(\mathcal{P})\| = \sqrt{\sum_i \boldsymbol{\varphi}_i^2(\mathcal{P})} = 1$ requires the map to have a unit Euclidean norm.

The Rarity Principle of Saliency. We refer to the objective of the optimisation problem (3.1) as the rarity principle of saliency. After rewriting it as $R(\boldsymbol{\varphi}(\mathcal{P})) = \sum_{i,j} \boldsymbol{\varphi}_i(\mathcal{P})\boldsymbol{\varphi}_j(\mathcal{P})\boldsymbol{M}_{ij}(\mathcal{P})$ and assuming that the saliency map $\boldsymbol{\varphi}(\mathcal{P})$ is binary (i.e. 1 for salient points and 0 otherwise), we can see that the objective exactly sums the shape contrasts among all sample points together. By globally optimising this combinatorial problem with the sparsity constraint, we would obtain a set of salient regions (associated with the salient points) that are the most distinct from others. However, this problem is known to be NP-hard to solve (Feige et al., 2001) and the resulting map is not continuous for many applications. Therefore, we relax a saliency map to be continuous-valued and arrive at our continuous metric-based rarity optimisation problem (3.1), so that it can be much more efficiently solved and the optimal saliency map is inherently continuous.

The Sparsity Principle of Saliency. We refer to the third constraint of the optimisation problem (3.1) as the sparsity principle of saliency. The constraint $\|\boldsymbol{\varphi}(\mathcal{P})\|_0 = \sum_i I[\boldsymbol{\varphi}_i(\mathcal{P}) \neq 0] \leq \mu N$ enforces the fraction of detected salient regions to be less than $0 < \mu \leq 1$, where $I[\cdot]$ is the indicator function. When $\mu = 1$, it has no use because all sample points are feasible to be identified as salient. When $0 < \mu < 1$, it guarantees that only a fraction of truly unique salient regions can be retained. We find that setting μ to 0.2 works well in practice. In the future work, we plan to continuously adjust this hyper-parameter from 1 to 0 to track the salient points that are persistent through the process, which can be identified as the true salient points.

To finely quantify and compare the sparsity patterns of saliency maps, we consider the Lorenz curves and Gini indices of them for analysis (Farris, 2010). Let $\boldsymbol{\varphi}_{(1)}(\mathcal{P}) \leq \boldsymbol{\varphi}_{(2)}(\mathcal{P}) \leq \cdots \leq \boldsymbol{\varphi}_{(N)}(\mathcal{P})$ be the non-decreasing order statistics of a saliency map $\boldsymbol{\varphi}$. The Lorenz curve is a piecewise linear function interpolating $N + 1$ points $(F_i, L(F_i))$, where for $0 \leq i \leq N$, $F_i = \frac{i}{N}$ denotes the proportion of the i least salient regions and $L(F_i) = \frac{\sum_{j=1}^{i} \boldsymbol{\varphi}_{(j)}(\mathcal{P})}{\sum_{j=1}^{N} \boldsymbol{\varphi}_{(j)}(\mathcal{P})}$ encodes the proportion of saliency values assigned to these regions. As F_i varies evenly from 0 to 1, $L(F_i)$ grows increasingly from 0 to 1, tracing out a concave curve from the origin to $(1, 1)$. The Gini index associated with a Lorenz curve is one minus two times the area under the curve. As shown in Fig. 3.3, the Lorenz curve of a uniform saliency map is the straight line from the origin to $(1, 1)$, with the lowest Gini index of 0 indicating the absolutely even distribution of saliency values to all regions. The other extreme is the singular saliency map, which distributes all the saliency values only to a single region and produces the highest Gini index of 1. We also show the saliency maps of the Dragon with and without sparsity in Fig. 3.3. It can be seen that the map without sparsity is visually and quantitatively very close to the uniform one, suggesting very weak discrimination between salient (i.e. the long body) and non-salient (i.e. the head and claws) regions of the Dragon. By enforcing the sparsity constraint in (3.1), we dramatically push the map away from the uniform one and highlight the salient regions of the Dragon much more clearly.

Multi-Scale Saliency Computation. To capture small- and large-scale salient regions, we use a metric for each scale to represent the global pairwise shape contrasts among all sample points for saliency computation. Specifically, we compute $\boldsymbol{M}_{ij}^{\tau}(\mathcal{P}) = \|f^{i,\tau} - f^{j,\tau}\|^2$ as the squared Euclidean distance between the descriptors of a pair of regions, and $\boldsymbol{M}^{\tau}(\mathcal{P}) \in \mathbb{R}^{N \times N}$ as the metric consisting of these descriptor distances among all points at scale τ. Due to the use of such

a global metric representation, we are able to avoid the ambiguity of manually choosing a suitable context for saliency detection, as traditionally done in the methods of (Lee et al., 2005; Gal and Cohen-Or, 2006; Feixas et al., 2009; Leifman et al., 2012; Wu et al., 2013; Pingping et al., 2015; Shilane and Funkhouser, 2007; Zhao et al., 2016; Jeong and Sim, 2017). More importantly, the representation is decoupled with the underlying mesh tessellation, which may contain topological flaws that prevent robust saliency computation.

Without the sparsity constraint in (3.1), the objective is the Rayleigh quotient of a metric $\boldsymbol{M}^{\tau}(\mathcal{P})$ and the saliency map $\boldsymbol{\varphi}^{\tau}(\mathcal{P})$ globally optimising it is the principal eigenvector of $\boldsymbol{M}^{\tau}(\mathcal{P})$ (Godsil et al., 2001). Due to the non-negativity of $\boldsymbol{M}^{\tau}(\mathcal{P})$, its principal eigenvector is guaranteed to be non-negative and unique. It can be efficiently computed from $\boldsymbol{M}^{\tau}(\mathcal{P})$ using the power method (Yuan and Zhang, 2013). With the sparsity constraint, we can also efficiently solve the problem (3.1) using the truncated power method (Yuan and Zhang, 2013). We describe the solution process as follows:

- **Initialisation**. We shift all the eigenvalues of $\boldsymbol{M}^{\tau}(\mathcal{P})$ to $(0, \infty)$ to make it positive definite, $\widetilde{\boldsymbol{M}^{\tau}(\mathcal{P})} \leftarrow \boldsymbol{M}^{\tau}(\mathcal{P}) + \nu \boldsymbol{I}$, where ν is the principal eigenvalue of $\boldsymbol{M}^{\tau}(\mathcal{P})$ computed from the power method and $\boldsymbol{I}$ is the identity matrix.

- **Iteration**. We start from the principal eigenvector of $\boldsymbol{M}^{\tau}(\mathcal{P})$ computed from the power method ,and then alternate between setting the $(1-\mu)N$ smallest values of the current map to zeros and multiplying it by $\widetilde{\boldsymbol{M}}^{\tau}(\mathcal{P})$ followed with normalisation, until converging to the optimal sparse saliency map.

Multi-Scale Saliency Integration To capture both small- and large-scale salient features, we integrate multi-scale shape information by summing the metrics of all scales together: $\boldsymbol{M}(\mathcal{P}) = \sum_{\tau} \boldsymbol{M}^{\tau}(\mathcal{P})$. We then compute the integrated saliency map $\boldsymbol{\varphi}(\mathcal{P})$ from $\boldsymbol{M}(\mathcal{P})$ by solving the problem (3.1) using the method described in the above. This way, we obtain a scale-free saliency map that fully adheres to the rarity and the sparsity principles of saliency. We also successfully avoid the cost of computing, summing, and then discarding multi-scale saliency maps as traditionally done in (Lee et al., 2005; Gal and Cohen-Or, 2006; Feixas et al., 2009; Leifman et al., 2012; Wu et al., 2013; Pingping et al., 2015; Song et al., 2014; Shilane and Funkhouser, 2007).

As shown in Fig. 3.4, the smallest-scale saliency map responds strongly to the local surface bumps

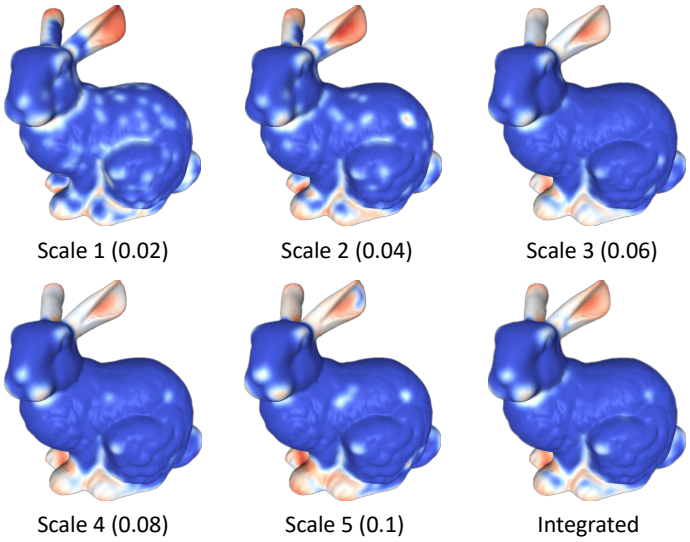


Figure 3.4: **Multi-scale Saliency Integration**. The saliency map of each scale is computed from the corresponding metric and the integrated map is computed from the sum of the metrics of all scales.

and textures of the Bunny. As the scale is increased, larger salient regions such as the mouth, eyes, ears, and feet are accurately captured. The final scale-free map effectively retains these visually salient regions while suppressing other undesirable local surface variations.

3.2.5 Vertex Saliency Interpolation

After computing the saliency values of sample points, we map them back to the underlying mesh vertices using Gaussian filtering. Let $\boldsymbol{\xi}_v$ denotes the saliency of a vertex v. We compute $\boldsymbol{\xi}_v$ as the Gaussian-weighted average of the saliency values of the sample points close to v:

$$\boldsymbol{\xi}_v = \frac{\sum_{i \in \mathcal{N}(v,3\sigma)} exp[-\left|\left|\boldsymbol{p}^v - \boldsymbol{p}^i\right|\right|^2/(2\sigma^2)]\boldsymbol{\varphi}_i(\mathcal{P})}{\sum_{i \in \mathcal{N}(v,3\sigma)} exp[-||\boldsymbol{p}^v - \boldsymbol{p}^i||^2/(2\sigma^2)]} \tag{3.2}$$

where $\mathcal{N}(v, 3\sigma) = \{i \mid \left|\left|\boldsymbol{p}^v - \boldsymbol{p}^i\right|\right| < 3\sigma\}$ and σ is the scale parameter of the Gaussian filter. We use a KD-tree to organise and query sample points for more efficient Gaussian filtering. We find that $\sigma = 0.02$ works well in practice.

3.3 Our 3D Eye Fixation Dataset

As visual saliency is inherently a pre-attentive mechanism of the human visual system (Borji and Itti, 2013), it is important to evaluate the performance of saliency detection methods using real

captured human eye movements on 3D surface meshes. However, the previously constructed saliency datasets are either too small (Howlett et al., 2005), only for 2D rendered images of 3D meshes (Kim et al., 2010), or not captured from real human eye movements on 3D meshes (Chen et al., 2012). Therefore, we propose our 3D eye fixation dataset for public saliency detection benchmarking, which is built as follows (see an example mesh and the collected eye fixations on the left of 3.7):

Mesh Dataset. We collected 50 meshes that are popularly used in computer graphics research from the Stanford 3D Scanning Repository (*The Stanford 3D scanning repository*, n.d.) and the SHREC2007 Challenge (*SHREC'2007 watertight mesh database*, n.d.). For each mesh, we fixed the non-manifold edges and remeshed the surface into good quality, so that all of our evaluated saliency detection methods can work well on it (Lee et al., 2005; Leifman et al., 2012; Pingping et al., 2015; Song et al., 2014). In the future, we will include meshes of poorer qualities (e.g. with holes and non-manifold structures) for more realistic benchmarking in real-world applications.

Participating Subjects. We hired 8 undergraduate and master students from Beihang University as human subjects for our study. They were aged 23-28 and have normal or corrected visions. They were kept unknown about the purpose of our study to reduce the bias of collected data.

Eye-Tracking Experiments. To capture 3D eye movement data, we generated a 48s video for each mesh that shows its whole surface from 12 key viewpoints. We kept each viewpoint static for 3s and then smoothly switched to the next viewpoint in 1s, so that the visual attention of a subject can be directed through the whole surface of a mesh. For each subject, we instructed him/her to sit in a distance of 95-110cm from a 1680×1050 LED display. Before the onset of each video stimulus (corresponding to each mesh in our dataset), we calibrated our used SMI RED250 eye-tracker by letting the subject gaze at 9 successive black dots on the screen. We considered the calibration successful if the gaze error was less than $0.8°$, otherwise we repeated the calibration process. After calibration, we let the subject freely view the displayed video of a mesh and used the eye-tracker to capture his/her gaze positions on the screen at 250HZ sampling rate, with a gaze capturing accuracy of $0.4°$.

Data Pre-processing and Aggregation. For each mesh in our dataset, we discarded the first two

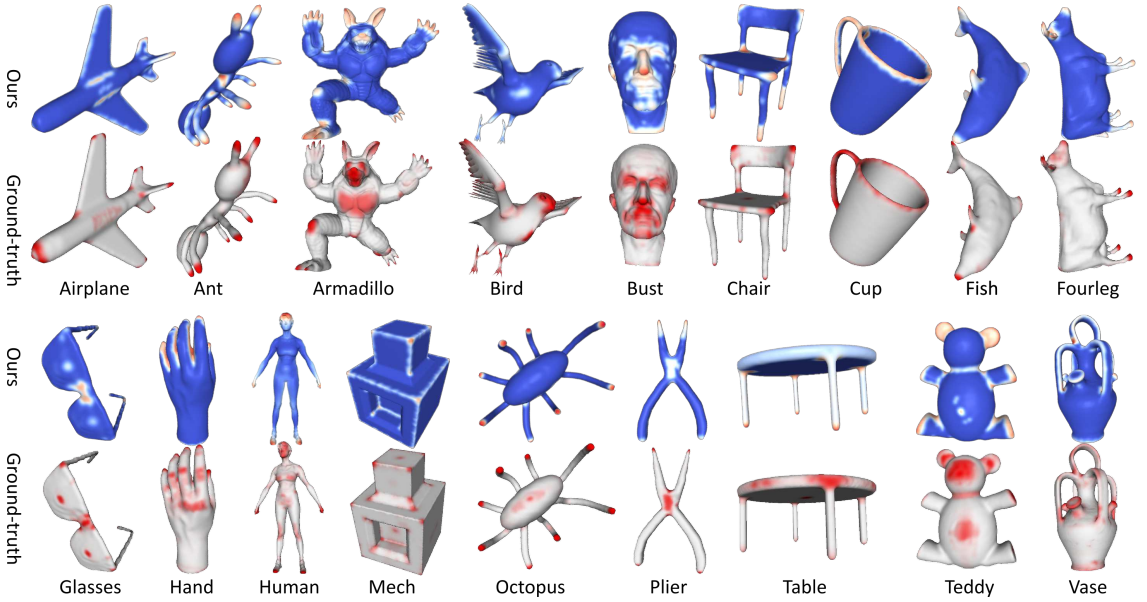


Figure 3.5: **Comparison of Our Saliency Maps with the Ground-truth of (Chen et al., 2012)**. Each of the 18 shown meshes belongs to a different object category from the dataset of (Chen et al., 2012). Following (Song et al., 2014), we take the rendered pseudo ground-truth images from (Chen et al., 2012) because we do not have access to their source code.

and the last two eye fixations of each subject because they can be noisy due to the onset and offset of the video stimulus. We only retained the 15 2D eye fixations of the longest duration (average $\geq$ 300ms) for each subject, because these eye fixations were much less noisy and represented where a subject was gazing towards on the screen. To obtain 3D eye fixations on the surface, we synchronised the timestamps of the captured 2D eye fixations with camera viewpoints and then projected them back to the nearest mesh vertices on the surface. Finally, we aggregated the 15 3D eye fixations from each of the 8 subjects to form 120 ground-truth eye fixations on each mesh surface in our dataset.

3.4 Results

3.4.1 Saliency Detection Results

We show the saliency maps of 18 meshes computed by our method in Fig. 3.5, along with the ground-truth maps provided by (Chen et al., 2012). Each mesh is randomly chosen from the 20 of the corresponding object category (Chen et al., 2012). These results indicate several strengths of our method for saliency detection:

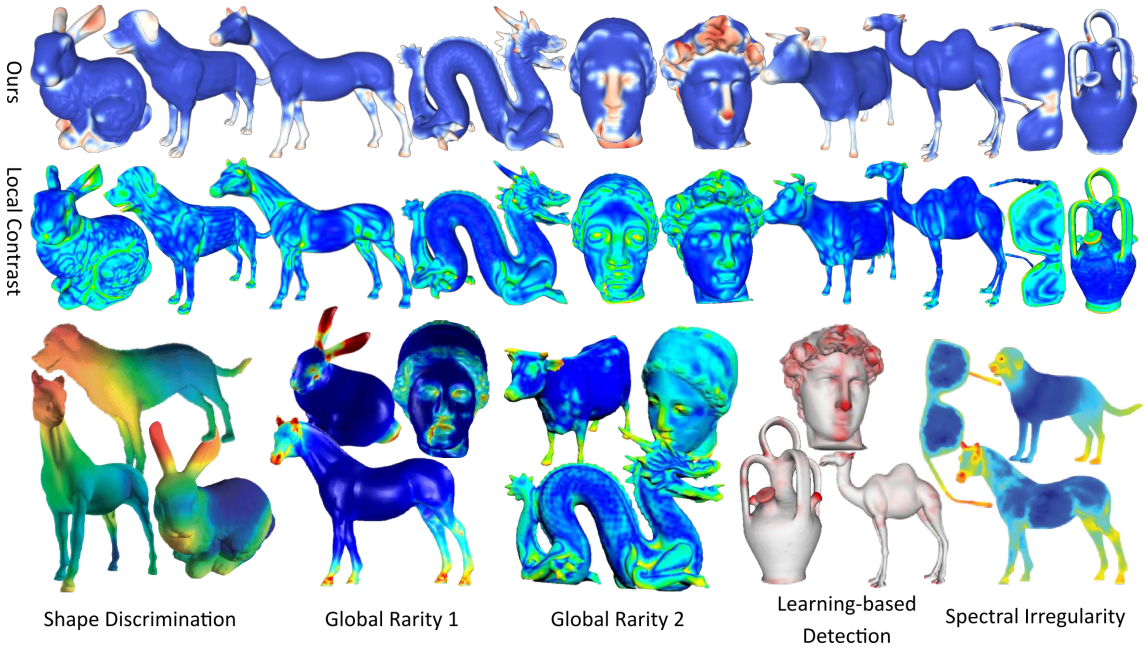


Figure 3.6: **Comparison of Our Saliency Maps with that Generated by Other Methods**. The competing methods include Local Contrast (Lee et al., 2005), Shape Discrimination (Shilane and Funkhouser, 2007), Global Rarity 1 (Leifman et al., 2012), Global Rarity 2 (Pingping et al., 2015), Learning-based Detection (Chen et al., 2012), and Spectral Irregularity (Song et al., 2014). Following (Leifman et al., 2012; Pingping et al., 2015; Song et al., 2014), we take the rendered saliency maps from the original papers of (Leifman et al., 2012; Pingping et al., 2015; Song et al., 2014; Shilane and Funkhouser, 2007; Chen et al., 2012) because we do not have access to their source codes. For the method of (Lee et al., 2005), we generate the saliency maps using our own implementation and visualise them to match the original colour themes.

- **Shape Distinction**. Our method successfully detects the globally distinct regions of surface meshes, such as the protruded parts (the horns, ears, and legs of the Cow), shape extremities (the hands, feet, and head of the Human), sharp edges (the perpendicular borders of the Mech), and corners (the claws of the Armadillo).
- **Curvature Insensitivity**. Our method is shown to be robust to the local curvature changes of surface regions. As shown for the legs of the Armadillo, they are bumpy and textured but are effectively suppressed by our method.
- **Compactness of Saliency**. The saliency maps computed by our method are visually quite compact, which only highlight a small number of salient regions with a clear boundary between non-salient ones.

We can also see that our saliency maps are visually close to the ground-truth. This suggests that they capture the true unknown human visual attention towards surface meshes to some extent. We note that our method fails to capture some ground-truth salient regions, such as the chest of Armadillo and the face of Teddy. This is because our method is purely unsupervised without using any semantic annotations for training or fine-tuning. In Chapter 4, we show that this can be addressed by learning metric from ground-truth saliency annotations.

3.4.2 Visual Comparisons with Other Methods

We compare our saliency maps with those generated by 6 representative methods in Fig. 3.6. We choose these methods for comparison because they are the most cited in the field and have distinct methodologies. We highlight the merits of our method over each of them as follows:

- **Local Contrast**. The method of (Lee et al., 2005) computes saliency as the local contrast of mean curvatures and is thus unable to suppress bumpy surface regions such as the body of the Bunny. In comparison, our method only detects the globally rare mouth, eyes, ears, and feet regions.
- **Shape Discrimination**. The method of (Shilane and Funkhouser, 2007) detects category-specific distinctive regions and only marks the whole heads of the Horse and the Dog as salient. Our method more finely captures the individual salient regions, including their mouths, eyes, ears, and legs.

- **Global Rarity**. While the method of (Leifman et al., 2012) used shape extreme points, patch distinction and patch association for saliency detection, the method of (Pingping et al., 2015) induced saliency from their dissimilarities to non-salient backgrounds. As shown for the facial features of the Horse and the Bunny, our method highlights them more accurately compared to that of (Leifman et al., 2012). For the body of the Dragon and the head of the Igea, our method is more robust to the noisy surface variations.

- **Learning-based Detection**. The method of (Chen et al., 2012) trains a tree-regression function for saliency detection. However, the trained function shows limited generalisation abilities for novel meshes. For example, the handles of the Vase and the legs of the Camel are not well detected. Our method captures these regions without using any semantic data.

- **Spectral Irregularity**. The method of (Song et al., 2014) leverages the residuals of the Laplacian spectrum for saliency detection. Due to the spatial unawareness of spectral basis, the method has difficulty in localising individual salient regions. While the forelegs of the Dog and the centres of the Glasses are of interest, they are erroneously neglected. Our method recovers them correctly.

3.4.3 Quantitative Comparison with Other Methods

We also evaluate saliency detection methods on our own 3D eye fixation dataset and the Schelling saliency dataset of (Chen et al., 2012). We choose the two datasets because the former is directly captured from our human eye tracking experiments and the latter reflects human subjective agreements on what constitute semantically prominent regions on surface meshes. Both characterise human visual attention on surface meshes to some extent.

Implementing Other Methods for Comparison. Since the introduction of mesh saliency to computer graphics (Lee et al., 2005), a number of saliency detection methods have been proposed in the past (Lee et al., 2005; Gal and Cohen-Or, 2006; Feixas et al., 2009; Zhao et al., 2016; Jeong and Sim, 2017; Shilane and Funkhouser, 2007; Chen et al., 2012; Leifman et al., 2012; Pingping et al., 2015; Wu et al., 2013), which show some progress of the visual quality of the generated saliency maps. However, their source codes are not publicly available, preventing a large-scale quantitative benchmarking of their true performance. (Kim et al., 2010) only evaluated the performance of the method of (Lee et al., 2005) and the evaluation is only based on 2D eye

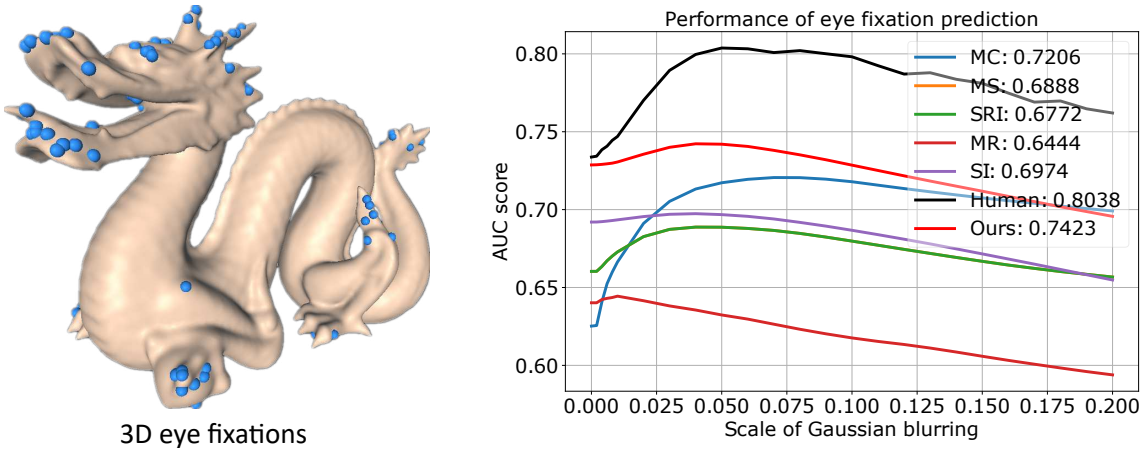


Figure 3.7: **Performance of Eye Fixation Prediction**. *Left*: The Dragon and our captured ground-truth 3D eye fixations on the surface. *Right*: The Human method uses the captured eye fixations of half subjects to predict that of the other halves, thereby measuring the self-consistency of our eye fixation dataset. The competing methods are MC (Meyer et al., 2002), MS (Lee et al., 2005), SRI (Leifman et al., 2012), MR (Pingping et al., 2015), and SI (Song et al., 2014). The peak AUC scores of these methods are displayed in the plot legend.

fixation data. Therefore, we have made efforts to implement the highly cited methods of (Lee et al., 2005; Leifman et al., 2012; Pingping et al., 2015) and use the source code of (Song et al., 2014) for quantitative evaluation. For the methods of (Lee et al., 2005; Song et al., 2014), we use the original parameters and find that the reproduced results align well with that reported in the original papers. For the methods of (Leifman et al., 2012; Pingping et al., 2015), we have tried to tune their shape descriptor construction and saliency computation parameters to match the reported images in the original papers. To our knowledge, this is the first time that saliency detection methods are quantitatively evaluated in the field. In the future, we plan to implement other methods for more thorough evaluation.

Evaluation on Our Eye Fixation Dataset. As described in Section 4, we build a 3D eye fixation dataset from 50 meshes and 8 human subjects, who were instructed to freely view these meshes on a computer screen while wearing a high-precision eye-tracking device. After data pre-processing and aggregation, we retain 120 most frequently attended points on each mesh in our study. We use the popular Area Under Curve (AUC) metric to quantify how well a saliency map captures these eye fixations on a surface (Borji and Itti, 2013).

We present an example mesh with the captured eye fixations and the evaluation results on our eye fixation dataset in Fig. 3.7. As pointed out by (Borji and Itti, 2013), the AUC metric is sensitive

to the blurring of saliency maps, so we filter each saliency map on a surface using the Gaussian kernels with scales from 0 to 0.2 and compute the corresponding AUC score for each scale. The Human method amounts to filtering the eye fixations of randomly selected half subjects into a saliency map and evaluating on the other half subjects for each mesh. A stable AUC score is computed by averaging the results of this random process.

It can be seen from Fig. 3.7 that the Human method detects the captured eye fixations very accurately, achieving the peak score around scale 0.05. This shows that our recorded eye fixations agree well among subjects and therefore qualify as ground-truth for method benchmarking. By focusing on the peak scores, it is surprising to see that MC performs better than MS on eye fixation prediction, which appears to contradict the findings of (Kim et al., 2010). This may be because Kim et al. (Kim et al., 2010) did not take blurring into consideration for evaluation. It is also interesting to see that the more global SRI and MR methods achieve lower accuracy compared to that by the local MC and MS methods, indicating their limited eye fixation localisation performance. The spectral SI method, compared to them, performs slightly better but is still worse than the optimally blurred MC method. In contrast, our method achieves much higher eye fixation localisation accuracy around the optimal blur scale 0.04.

Evaluation on the Saliency Dataset of (Chen et al., 2012). We also use the Schelling saliency dataset of (Chen et al., 2012) for method benchmarking. This dataset has 400 meshes evenly split into 20 object categories, with a collection of human-annotated salient points on each mesh and the corresponding filtered saliency map. We use the AUC score and the Linear Correlation Coefficient (LCC) to quantify the accuracy of a saliency map for predicting discrete salient points and continuous saliency values respectively (Borji and Itti, 2013). The LCC score of two saliency maps is computed by first normalising each map using its respective mean and standard deviation and then calculating the inner-product of the two normalised maps.

To finely compare different methods, we report their AUC scores for each object category separately and the scores for all categories together in Table 3.1. It can be seen that overall our method is the best performing one, and the followings are the MS, MC, SRI, SI, and MR methods. It is interesting to see that the AUC scores achieved by these methods are generally higher than that on our eye fixation dataset. This may be because the salient points in the dataset of (Chen et al., 2012) are fewer and less spread on mesh surfaces. The sparsity of salient points can explain the poor

Table 3.1: **The Performance (AUC) of Salient Point Detection on the Dataset of (Chen et al., 2012)**. The first row is the list of evaluated methods: MC (Meyer et al., 2002), MS (Lee et al., 2005), SRI (Leifman et al., 2012), MR (Pingping et al., 2015), SI (Song et al., 2014), and Ours. The second row shows the scores computed on all 20 categories of the dataset of (Chen et al., 2012) together. The remaining rows show the scores computed on each category separately. The highest score is highlighted in each row.

	MC	**MS**	**SRI**	**MR**	**SI**	**Ours**
All Categories	0.7839	0.8028	0.7605	0.6826	0.7097	**0.8168**
Airplane	0.8952	0.8356	**0.9004**	0.7404	0.8690	0.8790
Ant	0.7728	**0.8806**	0.7975	0.7331	0.7039	0.7925
Armadillo	0.8620	**0.9022**	0.7603	0.8165	0.7858	0.8878
Bearing	0.7814	0.8313	0.6350	0.6445	0.5312	**0.8555**
Bird	0.8442	0.7792	**0.8468**	0.7532	0.7878	0.8232
Bust	**0.8134**	0.8120	0.7696	0.6307	0.6714	0.7690
Chair	0.7570	0.7821	**0.8398**	0.6665	0.7154	0.8012
Cup	0.7845	0.7829	0.7888	0.5622	0.7891	**0.8031**
Fish	**0.9432**	0.9109	0.9231	0.8451	0.8950	0.9015
Fourleg	**0.8613**	0.8394	0.7682	0.7996	0.8281	0.8331
Glasses	0.5201	0.5981	**0.7057**	0.4969	0.5226	0.6947
Hand	0.7895	**0.8242**	0.8159	0.6977	0.7440	0.8066
Helix	**0.8771**	0.8702	0.6089	0.8025	0.8188	0.8462
Human	0.7515	**0.8015**	0.6661	0.6745	0.7085	0.6890
Mech	**0.8880**	0.8231	0.8060	0.6098	0.5934	0.8780
Octopus	0.7295	0.8339	0.7534	0.7063	0.6059	**0.8796**
Plier	0.5677	0.7083	0.8718	0.5119	0.6370	**0.9128**
Table	0.7734	0.7458	0.8123	0.7102	0.7355	**0.8317**
Teddy	0.6909	**0.7131**	0.5457	0.6160	0.6775	0.6975
Vase	0.7747	0.7822	0.7874	0.7201	0.7819	**0.8299**

performance of the SI and MR methods, which produce overly large patches of salient regions and therefore lack feature localisation ability. In contrast, our method localises salient points more accurately by optimising the rarity and sparsity principles together.

We report the LCC scores of these methods for predicting continuous saliency distributions in Table 3.2. We note that continuous saliency distributions are generally harder to predict than discrete points because a method needs to discriminate salient and non-salient regions more finely. Therefore, we expect the LCC metric to be a more comprehensive performance metric than AUC. It can be seen that our method produces saliency maps that correlate with the ground-truth considerably better compared to other methods. We observe that while MS and SRI may be good at localising sparse salient points, they have limited abilities to finely separate less salient from totally non-salient ones. As expected, SI and MR remain the worst-performing methods because they produce overly large patches of salient regions that contain many non-salient backgrounds as well.

Table 3.2: **The Performance (LCC) of Saliency Value Prediction on the Dataset of (Chen et al., 2012)**. The first row is the list of evaluated methods: MC (Meyer et al., 2002), MS (Lee et al., 2005), SRI (Leifman et al., 2012), MR (Pingping et al., 2015), SI (Song et al., 2014), and Ours. The second row shows the scores computed on all 20 categories of the dataset of (Chen et al., 2012) together. The remaining rows show the scores computed on each category separately. The highest score is highlighted in each row.

	MC	**MS**	**SRI**	**MR**	**SI**	**Ours**
All Categories	0.3442	0.3131	0.2898	0.2158	0.1987	**0.4303**
Airplane	0.4908	0.3270	0.4271	0.2221	0.3049	**0.6049**
Ant	0.3779	0.4648	0.3349	0.3324	0.2116	**0.6138**
Armadillo	0.4248	**0.4801**	0.1690	0.3096	0.2417	0.4658
Bearing	0.2949	0.3055	0.1835	0.0760	0.0213	**0.3578**
Bird	0.4594	0.3496	0.3738	0.2381	0.2371	**0.5319**
Bust	**0.3018**	0.2979	0.2614	0.1005	0.1577	0.2295
Chair	0.2484	0.2441	0.3574	0.2130	0.1803	**0.4618**
Cup	**0.4011**	0.3624	0.3789	0.2094	0.3455	0.3306
Fish	**0.5824**	0.4708	0.4412	0.3745	0.3499	0.5303
Fourleg	**0.4211**	0.2945	0.3004	0.2558	0.2794	0.4089
Glasses	0.1736	0.1499	0.2662	0.0680	0.1648	**0.3825**
Hand	0.3904	0.2714	0.3669	0.2806	0.2013	**0.4208**
Helix	0.4377	**0.4616**	0.0732	0.2528	0.2381	0.4448
Human	**0.3914**	0.3404	0.2579	0.2017	0.2020	0.2528
Mech	0.3862	0.2903	0.2166	-0.0082	0.0670	**0.4329**
Octopus	0.3029	0.3623	0.3319	0.1833	0.1048	**0.5367**
Plier	0.2799	0.1426	0.3719	0.3532	0.1330	**0.5450**
Table	0.3233	0.2315	0.3279	0.2484	0.2033	**0.5225**
Teddy	0.2137	**0.2419**	0.0942	0.1302	0.1466	0.2282
Vase	0.3810	0.3303	0.3160	0.2981	0.3300	**0.3830**

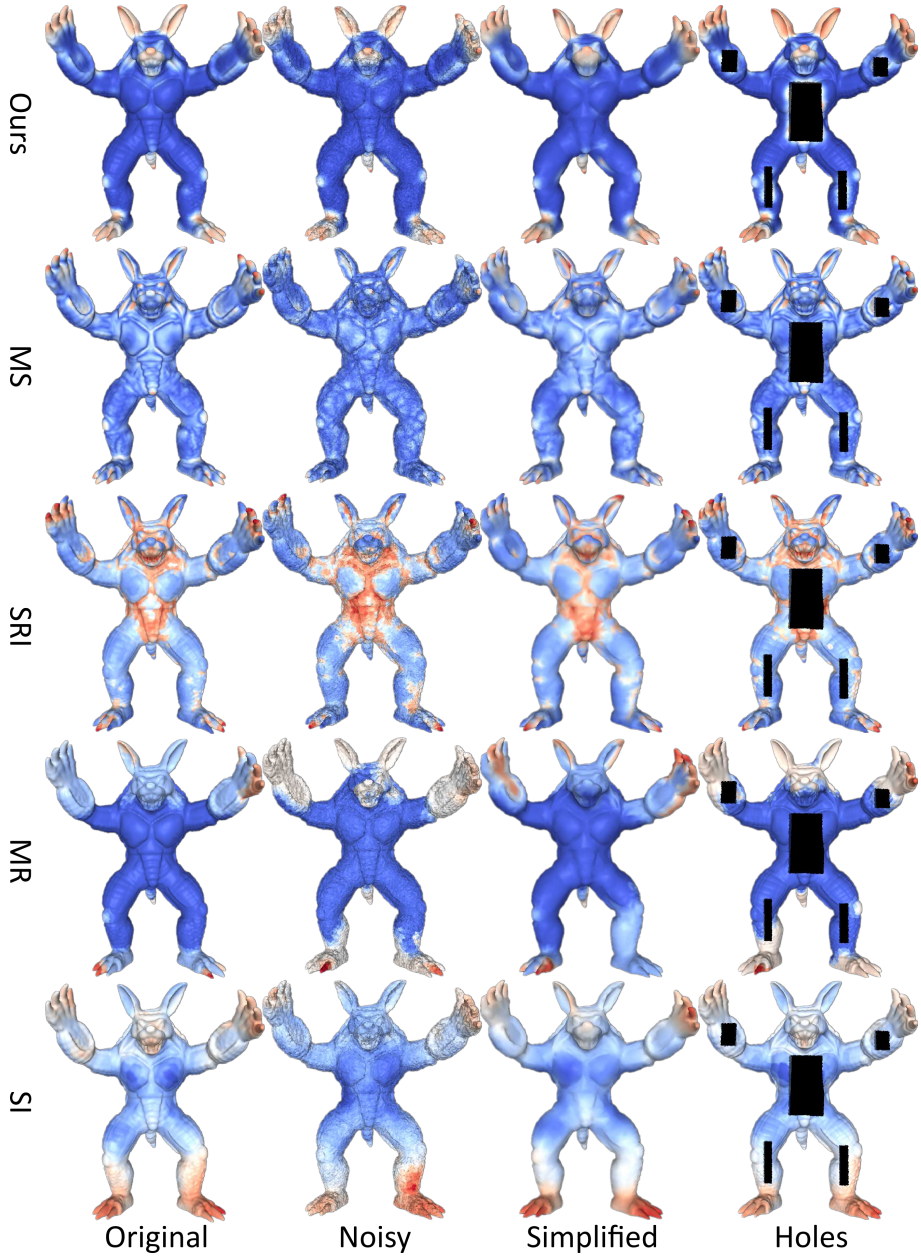


Figure 3.8: **Comparison of the Robustness of Our Method with that of Others**. The columns from left to right correspond to the original Armadillo mesh, the noisy version (20% noises in vertex normal directions), the simplified version (with 5k vertices), and the broken version with holes. The competing methods are MS (Lee et al., 2005), SRI (Leifman et al., 2012), MR (Pingping et al., 2015), and SI (Song et al., 2014).

3.4.4 Robustness Comparison with Other Methods

To show the robustness of our method, we compare our saliency maps of corrupted meshes with those computed by the competing methods of (Lee et al., 2005; Leifman et al., 2012; Pingping et al., 2015; Song et al., 2014) in Fig. 3.8. We add noises to the Armadillo mesh by randomly displacing the position of each vertex along the normal direction, with the displacement amount proportional to the unit distance. This generates roughly consistent noises on meshes when they are normalised to have the same radius. It can be seen that our method copes well with surface noises, simplifications, and holes. The computed saliency maps remain very close to that of the original Armadillo. In contrast, the method of (Lee et al., 2005) is fairly sensitive to curvature changes, responding strongly to the bumps around the legs of the Armadillo. The methods of (Leifman et al., 2012; Pingping et al., 2015) are also not sufficiently resilient to the introduced mesh flaws, producing inconsistent saliency maps for the damaged versions of the Armadillo (e.g. at the claws and facial regions). The method of (Song et al., 2014) appears more robust than that of (Leifman et al., 2012; Pingping et al., 2015) but fails to localise small-scale salient features such as the eyes, knees, and claws of the Armadillo.

3.4.5 Feature Points Localisation

To showcase the usefulness of our computed saliency maps, we apply them to the task of feature point localisation on surface meshes and evaluate on the dataset of (Dutagaci et al., 2012). We choose this task as our application because it is a fundamental building block of geometry processing and shape analysis.

The dataset of (Dutagaci et al., 2012) consists of 43 commonly used graphics meshes and the feature points annotated by 16 human subjects for each mesh. We evaluate six feature detectors: MS (Lee et al., 2005), SP (Castellani et al., 2008), SDC (Novatnack and Nishino, 2007), HKS (Sun et al., 2009), and ours. For MS, SP, SDC and HKS, Dutagaci et al. (Dutagaci et al., 2012) detected feature points using the published source codes. For our method, we classify a mesh vertex to be a candidate feature point if it has a local maxima saliency value that is also higher than the average of all local maxima saliency values. After sorting these candidates in the descending order of saliency, we sequentially retain each point with the constraint that it has at least 0.15 geodesic distances to the already selected points. This way, we obtain a set of feature points that are spread

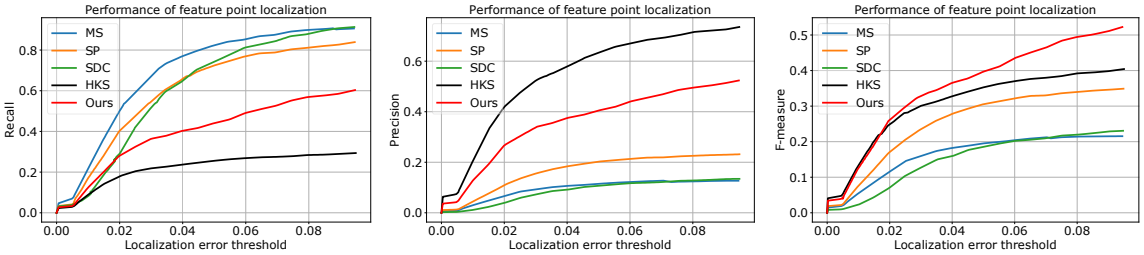


Figure 3.9: **Performance of Feature Point Localisation on the Dataset of (Dutagaci et al., 2012)**. Four competing methods that are highly cited in the field are included: MS (Lee et al., 2005), SP (Castellani et al., 2008), SDC (Novatnack and Nishino, 2007), and HKS (Sun et al., 2009). We use the popular Recall, Precision, and F-measure for performance evaluation (Dutagaci et al., 2012).

out on mesh surfaces. We use the popular Recall, Precision, and F-measure scores for detector performance evaluation (Dutagaci et al., 2012).

We show the Recall, Precision, and F-measure scores of the evaluated feature detectors in Fig. 3.9. The Recall score of a detector measures the fraction of true feature points it correctly finds in the ground-truth, and the Precision score measures the fraction of true feature points in the detector output. The F-measure score is the harmonic mean of Recall and Precision, which indicates the overall performance of a detector. As the localisation error threshold is increased, all three scores of the evaluated detectors grow because the output points become more probably to be identified as correct matches. The MS, SP, and SDC methods achieve higher Recalls compared to HKS and our method by detecting excessive numbers of points. The cost, however, is that they produce many non-salient points and therefore score considerably lower on Precision. HKS, on the other end, sacrifices Recall for Precision by only generating sufficiently prominent points. Our method is shown to strike the best balance between Recall and Precision, identifying many true feature points while not incurring many false positive ones. This can be seen from the F-measure plot, where our method is shown to outperform other feature detectors from very small localisation threshold.

3.4.6 The Run Times of Our Method

Table 3.3 reports the run times of our method on a commodity PC with a Dual Core 3.1GHZ CPU and a 4GB RAM. For each mesh, the run times of the main steps and the total time used are listed. It can be seen that our method scales well from medium-size meshes (e.g. the Bunny and the Dinosaur) to very large meshes (e.g. the Buddha and the Lucy). This high scalability

Table 3.3: **The Run Time of Our Method in Seconds**. **A**: multi-scale metric computation from a mesh. **B**: saliency computation from a metric. **C**: vertex saliency interpolation.

Mesh	#Vertices	A	B	C	Total
Bunny	35k	40.32	3.01	0.13	43.46
Dinosaur	56k	42.02	2.76	0.39	45.17
Armadillo	172k	40.45	3.05	0.75	44.25
Dragon	437k	41.39	3.03	1.58	46.00
Buddha	543k	40.47	2.87	1.92	45.26
Lucy	604k	43.34	2.94	2.57	48.85

would allow it to be used as an efficient preprocessing tool for many saliency-guided graphics applications.

3.5 Summary

In this chapter, we have proposed an accurate and robust sparse metric-based saliency detection method for 3D polygonal surface meshes. Our method was rigorously derived from optimising the rarity principle of saliency while enforcing the sparsity principle of saliency. This makes it able to optimally discover a compact set of salient regions that have the maximum distinction from others. Our method was formulated as solving for the sparse eigenvector of a global metric, which enjoys the robustness to the flaws of surface noises, simplifications, and holes. The results on our eye fixation dataset, the Schelling saliency dataset of (Chen et al., 2012), and the feature localisation dataset of (Dutagaci et al., 2012) show that our method produces more accurate saliency estimations compared with existing ones.

Chapter 4

Metric-based Unification of Saliency Detection and Non-rigid Shape Matching

In this chapter, we propose a unified metric representation framework for the joint analysis problems of saliency detection and non-rigid shape matching. This generalises the work in Chapter 3 that is rule-based shape analysis and only considers one task at a time. As shown in Fig. 4.1, the key idea of our approach is inferring saliency and matching from a single shared metric representation, which is learned from the input surface meshes using deep learning. This allows us to improve the generalisation of both tasks under non-rigid shape deformations, which we present in the following sections.

4.1 Introduction

The fundamental challenge of shape analysis is extracting knowledge from surface meshes that is not only understandable to humans but also invariant to complex shape deformations. Only with such invariance can a method work well consistently on the deformed versions of shapes. In this work, we narrow this challenge down to two fundamental shape analysis tasks: *mesh saliency detection* (Lee et al., 2005) and *non-rigid shape matching* (Van Kaick et al., 2011). We shall develop their underlying relation that is unknown to the field before, and exploit it for mutual

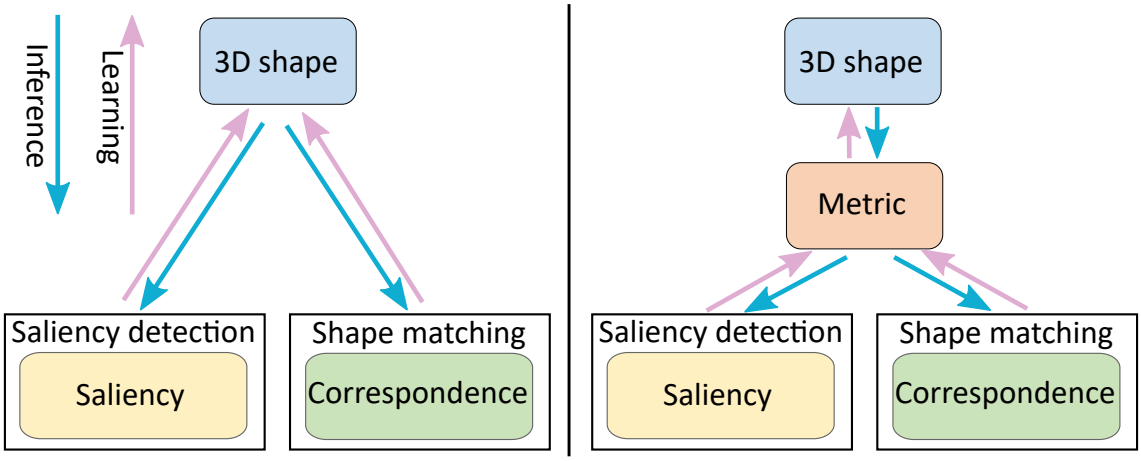


Figure 4.1: **The Overview of Our Unified Metric Framework for the Unification of Saliency Detection and Shape Matching**. While previous research approaches saliency detection and non-rigid shape matching separately and independently (*left*), we unifies them via a shared metric representation of surface meshes to better handle intra-category shape deformations for both sides (*right*).

improvements of saliency detection and shape matching using deep learning.

The first task we are interested in is saliency detection, which aims to compute a saliency map for an input mesh that signifies the perceptual or semantic importance of surface regions (Lee et al., 2005; Chen et al., 2012). Despite highlighting semantically important regions, saliency maps are also found to be consistent on surfaces of the same object category (Chen et al., 2012). However, the intra-category consistency of saliency has been ignored by the previous saliency detection methods (Lee et al., 2005; Chen et al., 2012; Song et al., 2014; Jeong and Sim, 2017) that compute saliency only for each mesh individually and separately without enforcing the consistency among meshes, which limits their generalisation abilities under complex intra-category shape deformations.

The other task we focus on is non-rigid shape matching, which finds semantically meaningful surface correspondences across meshes irrespective of the shape deformations among them (Van Kaick et al., 2011). As found in (Chen et al., 2012), human annotators tend to agree on a consistent set of semantically important regions on surfaces of the same object category, without communicating with each other during annotation. This shows that saliency is a strong deformation-invariant cue of shape matching within a category. However, existing shape matching methods mostly work on the matching task solely (Ovsjanikov et al., 2012; Kim et al., 2011; Boscaini, Masci, Rodolà and Bronstein, 2016), without exploiting the saliency cue for more robust

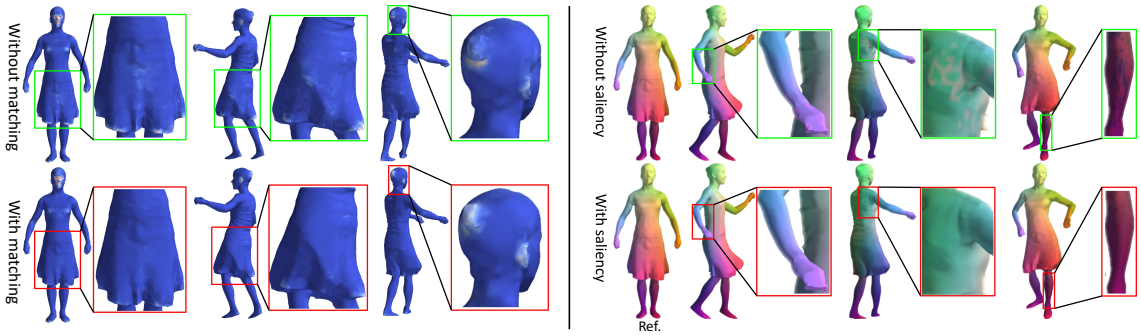


Figure 4.2: **The Mutual Benefits of Saliency and Matching**. Our method produces more deformation-invariant saliency maps with matching (*left*, using red and blue colours to visualise high and low saliency values respectively). It also produces more accurate shape matchings with saliency (*right*, colourising each target mesh vertex with its computed corresponding reference vertex's (X,Y,Z) coordinates).

matching.

In view of the existing gap between saliency and matching, we propose to unify the two tasks in the same framework so that they can help each other generalise better under complex intra-category shape deformations. To do this in a principled way, we need a unified representation of surface meshes that is geometry-aware, supports the joint modelling of saliency and matching, and most importantly enables the knowledge transfer between saliency and matching for mutual improvements. Towards this goal, we propose a unified metric representation that measures the pairwise semantic distances among all points on a mesh. Through two principled optimisation problems, we show that the saliency map and the shape embeddings of a mesh can be derived from the principal eigenvector and the smoothed Laplacian eigenvectors of the metric respectively (Fig. 4.1). Our joint modelling allows us to transfer the deformation-invariance (i.e. intra-category consistency) from matching to saliency for more accurate saliency detection. It also allows us to transfer the sparsity (i.e. semantic feature localisation) from saliency to matching for more robust correspondence solutions.

Having found a unified metric representation for saliency detection and shape matching, we need a way to compute the metric from the low-level geometry features of all points on an input mesh. More importantly, we wish the computation process to be differentiable so that it can be automatically learned from a given pair of saliency and matching datasets. Witnessing the success of deep neural networks for shape analysis (Boscaini, Masci, Rodolà and Bronstein, 2016), we propose a deep metric learning architecture that maps the low-level geometry features of all points on a mesh

to a semantics-aware metric representation for saliency detection and shape matching. The core of our architecture is a multi-layer RNN that can be learned to effectively integrate small-to-large scale shape features for each point. The other essential component of our architecture is a soft-thresholding operator, which can be learned to produce a sparse metric from the pooling result of the metrics computed from the RNN features of each scale. Our architecture is able to more effectively exploit multi-scale shape information and the sparsity of saliency, producing higher performance on saliency detection than alternatives.

To learn the deep metric representation from a pair of saliency and matching datasets, we propose a unified loss function with three terms: (1) the saliency fitting term to penalise the difference between the predicted and the ground-truth saliency maps of a mesh from the saliency dataset; (2) the saliency consistency term to penalise the difference between the predicted saliency maps of any pair of meshes from the matching dataset; (3) the metric consistency term to penalise the difference between the two metrics of any pair of meshes from the matching dataset. We minimise this loss function using our proposed eigenvector reparameterisation trick with the stochastic gradient descent (SGD) method (Bottou, 2010).

We jointly evaluate our method on saliency detection (Chen et al., 2012) and non-rigid shape matching (Anguelov et al., 2005; Vlasic et al., 2008; Bogo et al., 2014) datasets. The results show that it outperforms exiting rule-based and learning-based saliency detection methods in both the small and large sample training scenarios. It is also shown to improve both the model-based and learning-based methods for matching non-isometric pairs of shapes (Fig. 4.2). Our publicly available source code can be downloaded from this link*.

Our contributions include:

- We validate the mutual benefits between mesh saliency detection and non-rigid shape matching. Matching improves the accuracy and deformation-invariance of saliency via the intra-category consistency of matching, while saliency improves the robustness of matching under non-isometric deformations via the sparsity of saliency.
- We propose a unified metric representation for joint modelling of saliency and matching. The saliency map of a mesh is computed as the principal eigenvector of the metric and the

*https://drive.google.com/drive/folders/10Vu3ujF-5gPm8h_E35VhZR45WCjht18B

shape embeddings of the mesh are computed as the smoothed Laplacian eigenvectors of the metric. Our formulation allows matching to enforce the intra-category consistency for more accurate and deformation-invariant saliency detection, while exploiting the sparsity of saliency to induce semantically localised embeddings for more robust matching.

- We propose a multi-layer RNN architecture for more effectively integrating multi-scale shape information in metric computation, and an effective soft-thresholding operator for incorporating the sparsity of saliency in metric representation. We also propose a unified loss function for joint metric learning from a pair of saliency detection and shape matching datasets.

4.2 Our Unified Metric Representation

In this section, we propose a unified metric representation of surface meshes that enables the joint modelling of saliency detection and non-rigid shape matching. While multi-task learning is traditionally formulated as learning shared feature representations, it would be based on individual points on a surface and therefore lack a global geometry characterisation of the whole surface (Chen et al., 2012; Qi, Su, Mo and Guibas, 2017*a*; Qi, Yi, Su and Guibas, 2017; Yi et al., 2017). In contrast, we propose to represent the geometry of a mesh using a metric that characterises the pairwise learned distances among all points on the surface. We will show that such a global metric representation is essential to guaranteeing some desirable properties for saliency detection and shape matching.

4.2.1 Notations, Inputs, and Outputs

We denote a surface mesh as $\mathcal{P} = \{\boldsymbol{p}^k \in \mathbb{R}^3\}_{k=1}^N$ with N surface points. One quantity we want to compute for $\mathcal{P}$ is a non-negative-valued saliency map $\boldsymbol{\varphi}(\mathcal{P}) \in \mathbb{R}_{\geq 0}^N$, which assigns to each point $\boldsymbol{p}^k$ the saliency value $\boldsymbol{\varphi}_k(\mathcal{P})$. The higher the value, the more semantically important the point. The other quantity we want to compute is the shape embeddings $\boldsymbol{E}(\mathcal{P}) \in \mathbb{R}^{N \times m}$, which maps each 3D point $\boldsymbol{p}^k$ to a m-dimensional feature vector $\boldsymbol{E}_{k\cdot}(\mathcal{P})$ where non-rigid shape deformations can be simplified to rigid ones for more efficient matching (Maron et al., 2016). We denote the metric representation that leads to the two quantities as a non-negative-valued, symmetric, and zero-diagonal distance matrix $\boldsymbol{M}(\mathcal{P}) \in \mathbb{R}_{\geq 0}^{N \times N}$. It assigns a distance $\boldsymbol{M}_{ij}(\mathcal{P})$ to every pair of

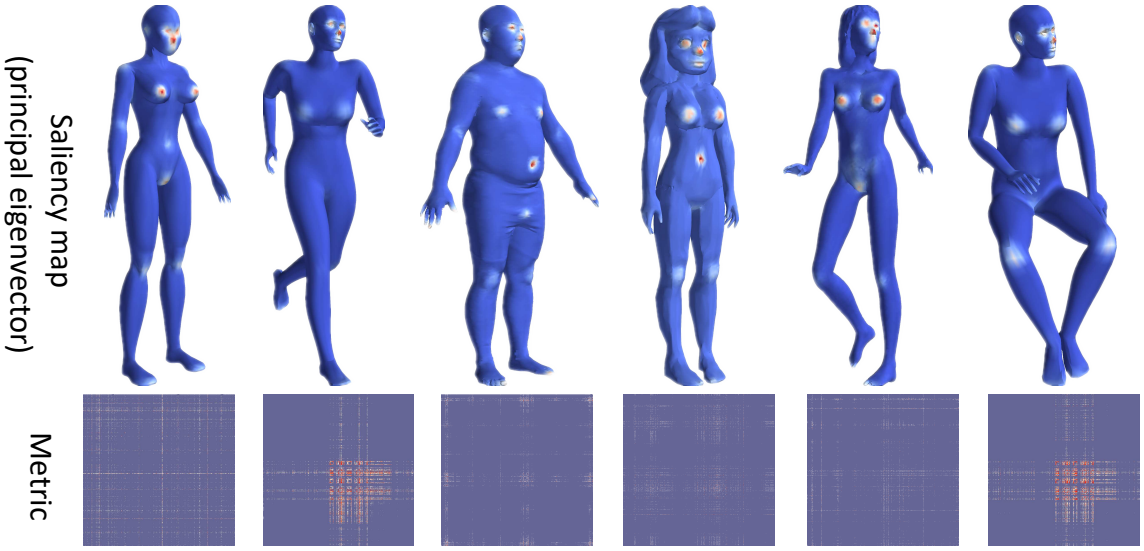


Figure 4.3: **The Sparsity of Saliency**. As human-annotated saliency maps only highlight a few semantically important regions on surfaces (Chen et al., 2012), our system automatically learns to produce sparse metrics whose principal eigenvectors (i.e. computed saliency maps) are sparse as well. Here, a redder matrix element represents a larger learned distance between the corresponding pair of surface points it visualises.

points $\boldsymbol{p}^i$ and $\boldsymbol{p}^j$ on the surface of $\mathcal{P}$.

In order to learn the metric $\boldsymbol{M}(\mathcal{P})$ for saliency detection and non-rigid shape matching, we require a pair of saliency and matching datasets, $\{\langle \mathcal{P}_i, \bar{\boldsymbol{\varphi}}(\mathcal{P}_i)\rangle\}_{i=1}^{N_s}$ and $\{\langle \mathcal{P}_i, \mathcal{P}_i'\rangle\}_{i=1}^{N_c}$, for training. In the former, each mesh $\mathcal{P}_i$ has the ground-truth saliency map $\bar{\boldsymbol{\varphi}}(\mathcal{P}_i)$. In the latter, every pair of meshes $\mathcal{P}_i$ and $\mathcal{P}_i'$ have a natural one-to-one semantic correspondence between their surface points.

As will be elaborated in Section 4.3, we use a stack of RNNs and a soft-thresholding operator to compute the metric from raw shape features. Therefore, the trainable parameters of our system include the RNN parameters and the threshold parameter. We train the system end-to-end using the SGD method (Bottou, 2010).

4.2.2 Saliency Detection from a Metric

In this subsection, we propose a differentiable saliency definition based on the metric representation of a mesh. We formulate the saliency map of a mesh as the global optimal solution to a metric-based optimisation problem, obtaining the solution as the principal eigenvector of the metric. This solution is differentiable and thus learnable, guarantees the non-negativity of saliency, and inherently encodes the sparsity of saliency for saliency detection and shape matching.

To begin with, we first consider $\boldsymbol{\varphi}(\mathcal{P})$ as a binary saliency map: $\boldsymbol{\varphi}_k(\mathcal{P}) = 1$ if $\boldsymbol{p}^k$ is a salient point and $\boldsymbol{\varphi}_k(\mathcal{P}) = 0$ otherwise. We then consider the problem of labelling a set of salient points so that the sum of their mutual distances, $\boldsymbol{\varphi}^T \boldsymbol{M}(\mathcal{P})\boldsymbol{\varphi} = \sum_i \sum_j \boldsymbol{\varphi}_i \boldsymbol{\varphi}_j \boldsymbol{M}_{ij}(\mathcal{P})$, can be maximised. Finally, since solving this problem is difficult and only produces a binary saliency map, we relax it as follows by replacing the binary saliency labels with continuous saliency values:

$$\boldsymbol{\varphi}(\mathcal{P}) = \arg\max \boldsymbol{\varphi}^T \boldsymbol{M}(\mathcal{P})\boldsymbol{\varphi}, \text{ s.t. } \boldsymbol{\varphi} \geq 0 \text{ and } \|\boldsymbol{\varphi}\|_2 = 1, \tag{4.1}$$

where we enforce the unit Euclidean norm[†] constraint for solution well-posedness. Without the non-negativity constraint, the objective of the problem is known as the Rayleigh quotient of the metric $\boldsymbol{M}(\mathcal{P})$ and the solution that globally maximises it is the principal eigenvector of $\boldsymbol{M}(\mathcal{P})$. Since the metric is symmetric and non-negative-valued by definition, its principal eigenvector is unique and guaranteed to be non-negative-valued according to the Perron-Frobenius theorem (Berman and Plemmons, 1994; Harel et al., 2006). Therefore, the optimal saliency map $\boldsymbol{\varphi}(\mathcal{P})$ is the principal eigenvector of the metric $\boldsymbol{M}(\mathcal{P})$.

Compared to existing saliency detection methods of (Lee et al., 2005; Leifman et al., 2012; Song et al., 2014; Pingping et al., 2015; Chen et al., 2012; Sinha et al., 2016; Qi, Su, Mo and Guibas, 2017*a*; Qi, Yi, Su and Guibas, 2017; Yi et al., 2017; Jeong and Sim, 2017), our metric-based saliency detection method has the following desirable properties:

- **Non-negative-Valued**. This is trivial but is not automatically satisfied by existing learning-based saliency detection methods, without the use of some non-linear activation functions that squash regression outcomes to non-negative saliency values. In contrast, our saliency map $\boldsymbol{\varphi}(\mathcal{P})$ is non-negative-valued by definition.

- **Differentiable**. As the metric $\boldsymbol{M}(\mathcal{P})$ is symmetric, $\boldsymbol{\varphi}(\mathcal{P})$ being one of its eigenvectors is continuously differentiable with respect to it (Berman and Plemmons, 1994). This allows us to fit $\boldsymbol{\varphi}(\mathcal{P})$ to the ground-truth saliency map $\bar{\boldsymbol{\varphi}}(\mathcal{P})$ and the map of any corresponding mesh $\mathcal{P}'$, producing more accurate and deformation-invariant saliency maps than existing rule- and learning-based methods.

- **Encoding Sparsity**. Apart from the intra-category consistency, the other characteristic of

[†] $\|\boldsymbol{x}\|_2 = \sqrt{\sum_i \boldsymbol{x}_i^2}$

the Schelling saliency maps is that they are sparse (Chen et al., 2012). When fitted to them, our saliency map $\boldsymbol{\varphi}(\mathcal{P})$ becomes sparse as well and drives a large fraction of the entries of the metric $\boldsymbol{M}(\mathcal{P})$ to zeros, which encode distances among non-salient points (Fig. 4.3). This sparsification mechanism is key to deriving semantically localised shape embeddings for more robust shape matching (4.2.3).

4.2.3 Non-rigid Shape Matching from a Metric

Having formulated the saliency map as the principal eigenvector of a metric in Section 4.2.2, we now describe how to obtain a shape embedding matrix $\boldsymbol{E}(\mathcal{P})$ from the same metric for robust shape matching. Our idea is to exploit the sparsity of saliency to learn better discrimination for salient points and more invariance for non-salient points. To do this, we formulate $\boldsymbol{E}(\mathcal{P})$ as the Laplacian embeddings with the metric $\boldsymbol{M}(\mathcal{P})$ and the surface connectivity of a mesh, so that they can be smooth, orthogonal, semantically localised, and deformation-invariant.

Following the setting of (Maron et al., 2016), we aim at computing a set of discriminative and deformation-invariant embedding coordinates $\boldsymbol{E}_{k\cdot}(\mathcal{P})$ for each surface point $\boldsymbol{p}^k$, so that the non-rigid shape deformation between a pair of meshes $\mathcal{P}$ and $\mathcal{P}'$ in the original 3D space can be simplified to a rigid one in the higher-dimensional embedding space. While existing methods strive on the discrimination and invariance of shape embeddings (Corman et al., 2014; Litman and Bronstein, 2014; Boscaini, Masci, Rodolà, Bronstein and Cremers, 2016; Cosmo et al., 2016; Wei et al., 2016), they learn for each individual surface point separately and therefore cannot guarantee that the obtained embeddings are orthogonal or smooth. Moreover, they treat all points equally and ignore the fact that salient points are semantically more important and geometrically more consistent within a shape category (Chen et al., 2012).

To address these issues, we consider the following Laplacian embedding problem (Belkin and

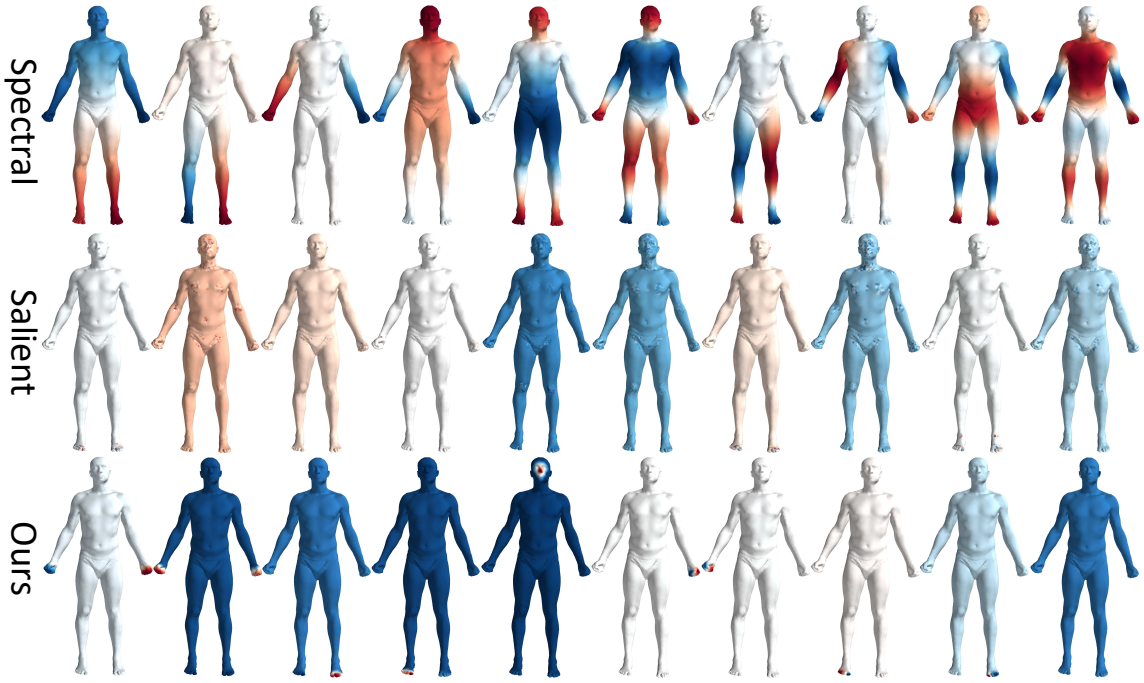


Figure 4.4: **Our Saliency-induced Embeddings**. The columns from left to right show individual embedding components computed by three different methods, with the colour visualising the smoothness and localisation of the embeddings on the surface. On top of being as locally smooth as the Laplacian spectral embeddings, our embeddings are further globally localised on the semantically important surface regions (i.e. eyes, ears, and limbs). Therefore, they are able to enforce additional constraints for robust shape matching.

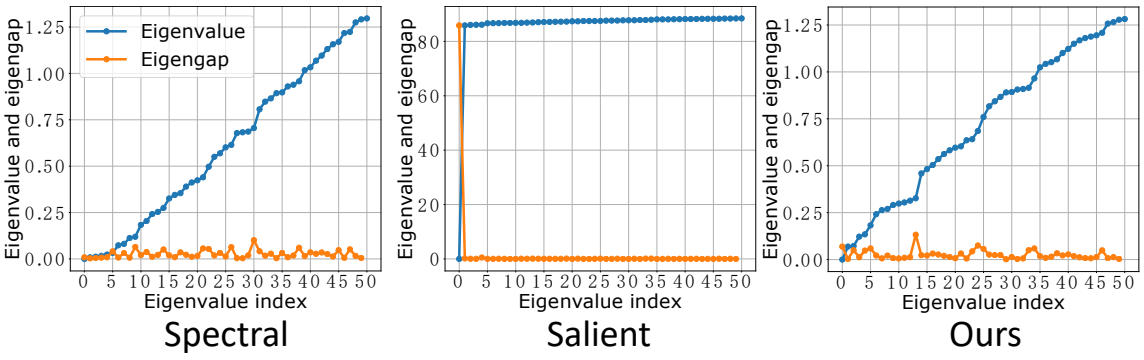


Figure 4.5: **The Deformation Stability of Our Embeddings**. Both the Laplacian spectral and our saliency-induced embeddings have non-zero eigengaps. Therefore, they can be made stable under complex intra-category shape deformations if the two metrics of any pair of meshes can be learned to be consistent within a shape category.

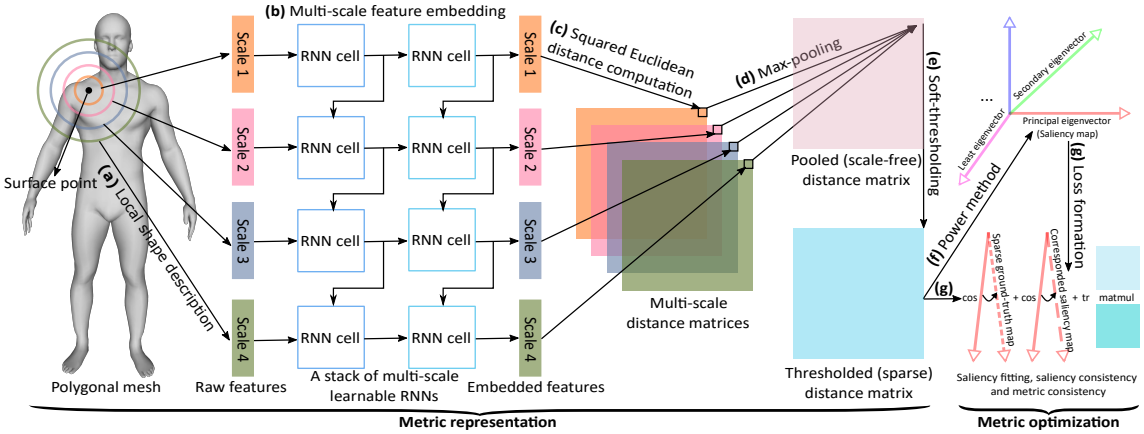


Figure 4.6: **The Overview of Our Deep Metric Learning Architecture**. The steps (**a**)-(**e**) are to compute a metric from the raw multi-scale features of a mesh and the steps (**f**)-(**g**) are to form the saliency fitting loss, saliency consistency loss, and metric consistency loss for metric learning. As our method uses a metric for joint modelling of saliency detection (principal eigenvector) and shape matching (Laplacian embeddings), it naturally incorporates the structure of all surface points for inference and learning.

Niyogi, 2002):

$$\boldsymbol{E}(\mathcal{P}) = \arg\min \operatorname{tr}[\boldsymbol{E}^T \Delta[\boldsymbol{A}(\mathcal{P})]\boldsymbol{E}], \tag{4.2a}$$

$$= \arg\min \underbrace{\frac{1}{2}\sum_{k=1}^{m}\sum_{i=1}^{N}\sum_{j=1}^{N}\boldsymbol{A}_{ij}(\mathcal{P})(\boldsymbol{E}_{ik}-\boldsymbol{E}_{jk})^2}_{\text{affinity-weighted smoothness penalties}}, \tag{4.2b}$$

$$\text{subject to } \underbrace{\boldsymbol{E} \perp \mathbf{1} \text{ and } \boldsymbol{E}^T\boldsymbol{E} = \boldsymbol{I}}_{\text{orthogonality constraints}}, \tag{4.2c}$$

where $\operatorname{tr}[\cdot]$ is the matrix trace[‡], $\Delta[\cdot]$ is the graph Laplacian, and $\boldsymbol{A}(\mathcal{P}) = (1-\theta)\boldsymbol{C}(\mathcal{P}) + \theta\boldsymbol{\varphi}(\mathcal{P})$ is a convex combination of the cotangent affinity matrix $\boldsymbol{C}(\mathcal{P})$ (Rustamov, 2007) and the salient affinity matrix $\boldsymbol{\varphi}(\mathcal{P})$ of a mesh. We compute $\boldsymbol{S}(\mathcal{P})$ by setting the diagonals of $1-\boldsymbol{M}(\mathcal{P})$ to zeros. While $\boldsymbol{C}(\mathcal{P})$ captures the affinities of adjacent surface points and $\boldsymbol{S}(\mathcal{P})$ encodes considerably large affinities among non-salient points, $\boldsymbol{A}(\mathcal{P})$ is a balance of them. As $\Delta[\boldsymbol{A}(\mathcal{P})]$ is symmetric and non-negative-definite, it is known that the optimal embeddings $\boldsymbol{E}(\mathcal{P})$ are its eigenvectors associated with the $m+1$ smallest eigenvalues (excluding the constant eigenvector corresponding to the eigenvalue of 0) (Belkin and Niyogi, 2002).

Compared with the shape embeddings of (Corman et al., 2014; Litman and Bronstein, 2014;

[‡] $\operatorname{tr}[\boldsymbol{X}] = \sum_i \boldsymbol{X}_{ii}$, $\Delta[\boldsymbol{X}]_{ii} = \sum_j \boldsymbol{X}_{ij}$ and $\Delta[\boldsymbol{X}]_{ij} = -\boldsymbol{X}_{ij}$ if $i \neq j$

Boscaini, Masci, Rodolà, Bronstein and Cremers, 2016; Cosmo et al., 2016; Wei et al., 2016), our saliency-induced ones have the following desirable properties:

- **Orthogonal**. Because $\Delta[\boldsymbol{A}(\mathcal{P})]$ is symmetric, the embedding coordinates $\boldsymbol{E}(\mathcal{P})$ being m of its eigenvectors are orthogonal to each other by definition. Therefore, our shape embeddings are mutually uncorrelated as the Laplacian spectral embeddings (Rustamov, 2007).

- **Smooth**. When setting θ to 0, we recover the Laplacian spectral embeddings of a mesh from (4.2a,4.2b,4.2c), which are the smoothest orthogonal functions on the surface (Rustamov, 2007) (Fig. 4.4, top). By setting θ to 0.1 to account for the affinities of adjacent surface points, we are able to ensure that our embeddings are smooth and orthogonal at the same time (Fig. 4.4, bottom).

- **Semantically Localised**. When setting θ to 1, we obtain embeddings that are localised on salient points (Fig. 4.4, middle), as the embedding smoothness among non-salient points is heavily enforced due to their much larger learned mutual affinities. Empirically, by setting θ to 0.1, we obtain both smooth and semantically localised embeddings (Fig. 4.4, bottom). Setting θ to a larger or a smaller value would weaken the smoothness or the localisation property.

- **Deformation-Invariant**. According to the Davis-Kahan theorem described in (Von Luxburg, 2007), we have the following bound on the distance between the shape embeddings of two meshes:

$$\mathrm{d}(\boldsymbol{E}(\mathcal{P}), \boldsymbol{E}(\mathcal{P}')) \leq \frac{\|\Delta[\boldsymbol{A}(\mathcal{P})] - \Delta[\boldsymbol{A}(\mathcal{P}')]\|_F}{\lambda_{m+1} - \lambda_m}, \tag{4.3}$$

  where $\mathrm{d}(\cdot, \cdot)$ is the Euclidean norm of the sines of the principal angles between $\boldsymbol{E}(\mathcal{P})$ and $\boldsymbol{E}(\mathcal{P}')$, and $0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$ are the non-decreasing eigenvalues of $\Delta[\boldsymbol{A}(\mathcal{P})]$. To lower this bound, we need to decrease its numerator by enforcing the deformation-invariance of the pair of learned metrics $\boldsymbol{M}(\mathcal{P})$ and $\boldsymbol{M}(\mathcal{P}')$. We also set $\theta = 0.1$ to ensure its denominator (the eigengap) is non-negligible, preventing divergence of the bound (Fig. 4.5). This ensures that our embeddings are sufficiently deformation-invariant for shape matching.

4.3 Our Deep Metric Learning Architecture

In Section 4.2, we have proposed a unified metric representation of surface meshes whose principal eigenvector and smoothed Laplacian eigenvectors can be used for saliency detection and non-rigid shape matching respectively. In this section, we propose a deep neural network architecture for computing the metric from an input mesh. The reason we need a deep architecture is that it is learnable and sufficiently powerful to extract high-level features from low-level geometry data for shape analysis (Boscaini, Masci, Rodolà and Bronstein, 2016; Litany et al., 2017; Qi, Su, Mo and Guibas, 2017*a*; Yi et al., 2017). As shown in Fig. 4.6, for each point on a surface mesh, we first (a) extract a set of raw multi-scale feature vectors and then (b) feed them into our proposed multi-layer RNN for multi-scale feature embedding. We then (c) compute a set of multi-scale Euclidean metrics to (d) derive a scale-free metric via max-pooling. Afterwards, we (e) use our proposed soft-thresholding operator to adaptively sparsify this metric and (f) compute the principal eigenvector to (g) form three loss terms. Finally, we minimise these terms together using our proposed eigenvector reparameterization trick with the SGD method (Bottou, 2010).

4.3.1 Our RNN for Multi-scale Feature Embedding

In this section, we describe our RNN method for multi-scale feature embedding. The inputs to our method are the raw multi-scale shape descriptors of a mesh $\mathcal{P}$, $\{\boldsymbol{F}^{\tau,0}(\mathcal{P}) \in \mathbb{R}^{N\times d}\}_{\tau=1}^{N_\tau}$, where N_τ is the number of scales from small to large and d is the feature dimension of each surface point at each scale τ. The outputs produced by our method are the embedded multi-scale features $\{\boldsymbol{F}^{\tau}(\mathcal{P}) \in \mathbb{R}^{N\times d}\}_{\tau=1}^{N_\tau}$, which are used for subsequent metric computations. As the shape information of each surface point naturally spans increasingly larger contexts and these contexts are not independent of each other (Sun et al., 2009; Kazhdan et al., 2003), it is difficult for some hand-crafted rules to discover the optimal correlation among multiple contexts and integrate them effectively (Qi, Yi, Su and Guibas, 2017). This motivates us to consider the multi-step RNN architecture that is usually popular for temporal sequence modelling.

Our idea is to learn features in two directions: the vertical direction that maps features from one layer to the next and the horizontal direction that propagates features from one scale to the next. More specifically, we propose to order shape features from small to large scales and then treat each scale as one step of an RNN in the scale sequence (Fig. 4.7, bottom right). This allows us to

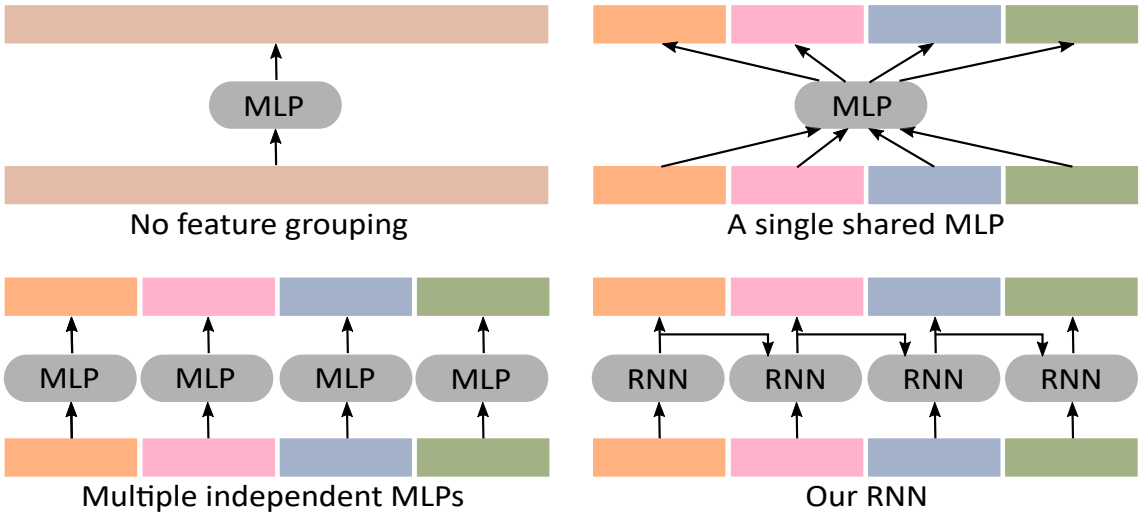


Figure 4.7: **Multi-scale Feature Embedding Architectures**. The image shows three baselines and our RNN method for multi-scale feature embedding. A single MLP can transform the concatenated features of all scales jointly (*top left*) or the features of each scale individually (*top right*), and multiple MLPs can work on each scale separately with no feature sharing among each other (*bottom left*). In contrast, our RNN method works on a sequence of small-to-large scale features and explicitly learns the transition between scales for more effective scale integration (*bottom right*).

parameterise our feature embedding architecture as a multi-layer function $\boldsymbol{f} = \boldsymbol{f}^{N_l} \circ \cdots \circ \boldsymbol{f}^{1}$, each layer of which is an RNN with our specially designed scale interpolation cell structure as follows:

$$\boldsymbol{O}^{\tau,l} = \underbrace{\tanh[\boldsymbol{\Upsilon}(\boldsymbol{F}^{\tau-1,l}\boldsymbol{W}^{\diamond,l} + \boldsymbol{F}^{\tau,l-1}\boldsymbol{M}^{\diamond,l})]}_{\text{predicting candidate output features}}, \tag{4.4a}$$

$$\boldsymbol{P}^{\tau,l} = \underbrace{\text{sigmoid}[\boldsymbol{\Upsilon}(\boldsymbol{F}^{\tau-1,l}\boldsymbol{W}^{\circ,l} + \boldsymbol{F}^{\tau,l-1}\boldsymbol{M}^{\circ,l})]}_{\text{predicting scale interpolation weights}}, \tag{4.4b}$$

$$\boldsymbol{F}^{\tau,l} = \underbrace{\boldsymbol{\Upsilon}[(1-\boldsymbol{P}^{\tau,l}) \odot \boldsymbol{F}^{\tau-1,l} + \boldsymbol{P}^{\tau,l} \odot \boldsymbol{\Upsilon}(\boldsymbol{O}^{\tau,l})]}_{\text{interpolating features via convex combination}}, \tag{4.4c}$$

where l is the layer of each RNN, $\{\boldsymbol{W}^{\diamond,l}, \boldsymbol{M}^{\diamond,l}\}_{l=1}^{N_l}$ are the learnable matrix parameters of the RNN, and $\boldsymbol{\Upsilon}(\cdot)$ is the feature-wise standardisation operator (Ioffe and Szegedy, 2015). To our knowledge, this is the first time that RNNs are used for multi-scale feature learning in shape analysis.

Compared with the alternatives shown in Fig. 4.7, our RNN learns scale integration explicitly to yield more powerful multi-scale features for shape analysis. Compared with long short-term

memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014), our cell has a simpler and more effective scale integration mechanism for multi-scale feature embedding.

4.3.2 Our Soft-thresholding Operator for Metric Sparsification

Human-annotated saliency maps only highlight a few semantically important regions on surfaces (Chen et al., 2012). However, existing saliency detection methods of (Lee et al., 2005; Leifman et al., 2012; Song et al., 2014; Pingping et al., 2015; Chen et al., 2012) do not enforce the sparsity of saliency, producing excessive amounts of regions that are actually not salient (Fig. 4.14). This motivates us to directly incorporate the sparsity of saliency into metric representation for more accurate saliency detection (Fig. 4.8). Our idea is to adaptively soft-threshold a metric using a parametric threshold learned from ground-truth saliency maps. We propose our soft-thresholding operator as follows:

$$\boldsymbol{M}(\mathcal{P}) = \max\{\dot{\boldsymbol{M}}(\mathcal{P}) - \Theta_t, 0\}, \tag{4.5}$$

where $\dot{\boldsymbol{M}}(\mathcal{P})$ is the scale-free metric computed via max-pooling described below, and Θ_t is a scalar parameter that can be learned to truncate the small elements of $\dot{\boldsymbol{M}}(\mathcal{P})$ to exact zeros (see Section 4.3.3 for analysis). This way, we can learn to sparsify $\dot{\boldsymbol{M}}(\mathcal{P})$ based on ground-truth saliency maps and ensure that its derived saliency map $\boldsymbol{\varphi}(\mathcal{P})$ is properly sparsified as well.

We now describe how to compute $\dot{\boldsymbol{M}}(\mathcal{P})$. From the embedded features $\boldsymbol{F}^\tau(\mathcal{P})$ of each scale τ, we can compute a squared Euclidean distance matrix $\ddot{\boldsymbol{M}}^\tau(\mathcal{P})$ among the N points of a mesh (Dokmanic et al., 2015). We choose this representation because it is simple, differentiable, and analytically computable via fast matrix and vector operations. To address the rank-deficiency of a single Euclidean metric (Dokmanic et al., 2015), we compute a scale-free metric by max-pooling the Euclidean metrics of all scales, $\dot{\boldsymbol{M}}(\mathcal{P}) = \max\{\ddot{\boldsymbol{M}}^1(\mathcal{P}), \ddot{\boldsymbol{M}}^2(\mathcal{P}), \ldots, \ddot{\boldsymbol{M}}^{N_\tau}(\mathcal{P})\}$, where the output is no longer low-rank as the linear independence of its rows (or columns) is greatly strengthened by the non-linear element-wise pooling operation.

Compared with traditional methods that enforce sparsity via a sparsity-inducing norm (Yuan and

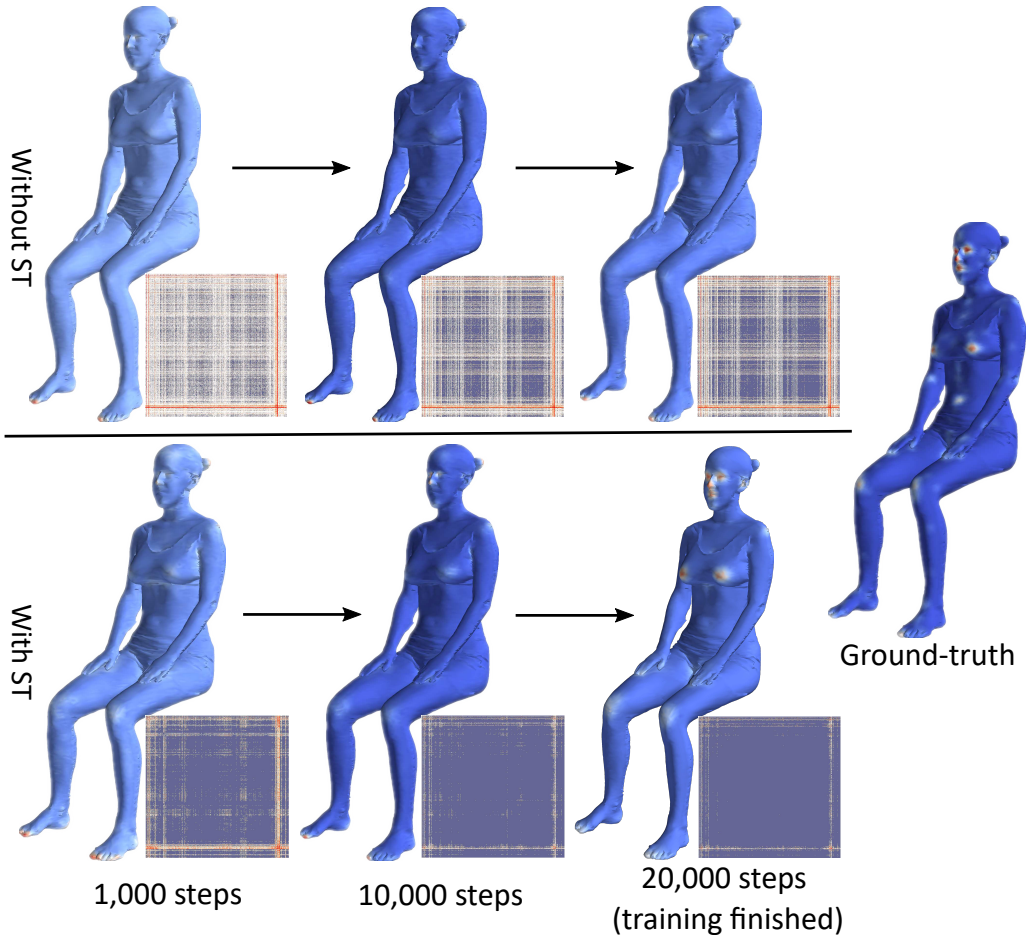


Figure 4.8: **The Effect of Our Soft-thresholding Operator**. We learn a soft-thresholding (ST) operator to adaptively truncate the small elements of a metric to exact zeros, improving the sparsity and accuracy of computed saliency maps significantly.

Zhang, 2013), ours learns sparsity by optimising Θ_t adaptively, without the need of weighting a sparsity-inducing norm by trial-and-error. This leads to much more accurate and sparser saliency maps (Fig. 4.8).

4.3.3 Our Multi-objective Loss Function for Metric Learning

Finally, we propose a loss function for metric learning from a given pair of saliency and matching datasets:

$$\mathcal{L}(\mathcal{P}, \mathcal{P}') = \alpha\mathcal{L}_\alpha(\mathcal{P}) + \beta\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}') + \gamma\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}'), \tag{4.6a}$$

$$\mathcal{L}_\alpha(\mathcal{P}) = 1 - \boldsymbol{\varphi}(\mathcal{P})^T\bar{\boldsymbol{\varphi}}(\mathcal{P}), \tag{4.6b}$$

$$\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}') = 1 - \boldsymbol{\varphi}(\mathcal{P})^T\boldsymbol{\varphi}(\mathcal{P}'), \tag{4.6c}$$

$$\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}') = 1 - \mathrm{tr}[\boldsymbol{M}(\mathcal{P})\boldsymbol{M}(\mathcal{P}')], \tag{4.6d}$$

where the *saliency fitting term* $\mathcal{L}_\alpha(\mathcal{P})$ penalises the difference between the predicted and ground-truth saliency maps of a mesh from the saliency dataset, the *saliency consistency term* $\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}')$ penalises the difference between the predicted saliency maps of any pair of meshes from the matching dataset, and the *metric consistency term* $\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}')$ penalises the difference between the two metrics computed from any pair of meshes from the matching datasets. α, β, and γ are their respective weights.

Our Eigenvector Reparameterisation. As the derivatives of $\boldsymbol{\varphi}(\mathcal{P})$ with respect to $\boldsymbol{M}(\mathcal{P})$ require matrix pseudo-inverse (Magnus, 1985), $\mathcal{L}_\alpha(\mathcal{P})$ and $\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}')$ cannot be minimised directly. We tackle this by approximating $\boldsymbol{\varphi}(\mathcal{P})$ as follows:

$$\boldsymbol{\varphi}(\mathcal{P}) \approx \frac{\boldsymbol{M}(\mathcal{P})\tilde{\boldsymbol{\nu}}}{\tilde{\boldsymbol{\nu}}^T\boldsymbol{M}(\mathcal{P})\tilde{\boldsymbol{\nu}}}, \tag{4.7}$$

where $\tilde{\boldsymbol{\nu}}$ is a numerical version of $\boldsymbol{\varphi}(\mathcal{P})$ computed by the power method. This approximation holds because $\tilde{\boldsymbol{\nu}}$ is associated with the largest eigenvalue of $\boldsymbol{M}(\mathcal{P})$ and is thus orthogonal to the other eigenvectors. Compared with the low-order approximation of (Leordeanu et al., 2012), ours has a much simpler form and is significantly more accurate.

Our Saliency Fitting Term. To analyse learning dynamics, we insert (4.7) into (4.6b) to obtain

the partial derivatives of $\mathcal{L}_\alpha(\mathcal{P})$ with respect to $\boldsymbol{M}(\mathcal{P})$ and Θ_t:

$$\frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \boldsymbol{M}_{ij}(\mathcal{P})} = C_1 \tilde{\boldsymbol{\nu}}_i \tilde{\boldsymbol{\nu}}_j - C_2 \bar{\boldsymbol{\varphi}}_i(\mathcal{P}) \tilde{\boldsymbol{\nu}}_j, \tag{4.8a}$$

$$\frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \Theta_{\boldsymbol{t}}} = \sum_{i=1}^{N} \sum_{j=1}^{N} I\{\boldsymbol{M}_{ij}(\mathcal{P}) > \Theta_{\boldsymbol{t}}\} \frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \boldsymbol{M}_{ij}(\mathcal{P})}, \tag{4.8b}$$

where $0 < C_1 \leq C_2$ and $I\{\cdot\}$ is the indicator function. If the system wrongly predicts a low saliency value for $\boldsymbol{p}^i$, i.e. $\tilde{\boldsymbol{\nu}}_i < \bar{\boldsymbol{\varphi}}_i(\mathcal{P})$, it will increase the distances of $\boldsymbol{p}^i$ to the other points because the derivatives $\{\frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \boldsymbol{M}_{ij}(\mathcal{P})}\}_{j=1}^{N}$ are negative. Conversely, its distances to the other points will decrease. Sparse ground-truth saliency maps therefore lead to the sparsification of $\boldsymbol{M}(\mathcal{P})$. Because $\frac{\partial \mathcal{L}_\alpha(\mathcal{P})}{\partial \Theta_{\boldsymbol{t}}}$ is the sum of mostly negative partial derivatives from pairs of salient points whose distances are large enough to exceed the threshold, $\Theta_{\boldsymbol{t}}$ increases during the training to drive the sparsification of $\boldsymbol{M}(\mathcal{P})$ further (Fig. 4.4 and 4.8).

Our Saliency Consistency Term. The form of $\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}')$ is the same as that of $\mathcal{L}_\alpha(\mathcal{P})$, except that we treat $\boldsymbol{\varphi}(\mathcal{P})$ and $\boldsymbol{\varphi}(\mathcal{P}')$ as each other's learning target. This allows us to enforce the intra-category consistency of saliency by pushing the predicted saliency maps of any pair of meshes closer to each other in a shape category.

Our Metric Consistency Term. To obtain deformation-invariant embeddings, we need to control the bound in (4.3) by minimising $\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}')$. This ensures that the learned metrics are sufficiently deformation-invariant. As the saliency consistency term $\mathcal{L}_\beta(\mathcal{P}, \mathcal{P}')$ can only regularise the principal eigenvector of the learned metric, we add the metric consistency term $\mathcal{L}_\gamma(\mathcal{P}, \mathcal{P}')$ to control the remaining eigenvectors.

4.4 Results

4.4.1 Implementation Details

We implement our proposed system in TensorFlow (V0.12). Throughout our experiments, we stack three layers of RNNs with an input and an output dimension of 256 each. We initialise the matrix parameters of each RNN to be orthogonal, and we initialise the soft-thresholding parameter to zero. We set the learning rates for the matrix and threshold parameters to 0.1 and 1×10^{-4} respectively, and decay them by a rate of 0.1 every 5,000 steps with a momentum of 0.9 for 20,000

SGD steps. We set the batch size to 1. At each training step, we randomly retrieve a mesh and its ground-truth saliency map from a saliency dataset, as well as a pair of meshes from a shape matching dataset. We resample each mesh to 500 surface points for efficient learning.

To learn a metric from a mesh, we use its spatial rather than spectral raw features, as the former capture both intrinsic and extrinsic geometry for shape analysis (Corman et al., 2017). Specifically, we use the spherical harmonic (SH) descriptors of (Kazhdan et al., 2003), which are derived from a raw distance field and have a theoretical guarantee of minimal information loss. We encode the local shape of each vertex with 16 SH amplitudes for each of 16 concentric shells of equally increasing radii, with the radius of the outmost shell being one-eighth of the mesh diameter. We pad the raw features of each scale with zeros to create a dimension of 256.

4.4.2 Evaluation of Our Deep Learning Architecture

In this section, we validate that our RNN method is more effective at learning multi-scale shape features compared with the baselines in Fig. 4.7, and that our soft-thresholding operator further improves the performance via adaptive metric sparsification. We train on an 80% random sample of the 20 meshes from each of the 20 categories of the Schelling saliency dataset (Chen et al., 2012) and test on the remaining meshes at each training step. Here, we use only the saliency fitting loss for large-scale evaluation because none of the 20 categories apart from the Human and Fourleg has corresponding shape matching datasets (Anguelov et al., 2005; Vlasic et al., 2008; Bogo et al., 2014). We use the Gini index to measure the sparsity of saliency maps and metrics (Hurley and Rickard, 2009).

Evaluation of Our RNN. First, we evaluate our RNN method for multi-scale feature learning. To match our architecture, we stack 3 layers of MLPs with an input and an output dimension of 256 for each of the three baselines in Fig. 4.7, and use the tanh activation function and feature standardisation for them. We find that the SGD parameters of our architecture work well for all of them as well. As shown in Fig. 4.9, neither a shared MLP nor multiple independent MLPs perform well, because the former ignores the feature characteristics of different scales and the latter fail to integrate features across scales. A single MLP performs much better as it transforms the features of all scales simultaneously. Still, our RNN achieves the lowest saliency testing error by explicitly learning scale transition and integration, both with and without the soft-thresholding

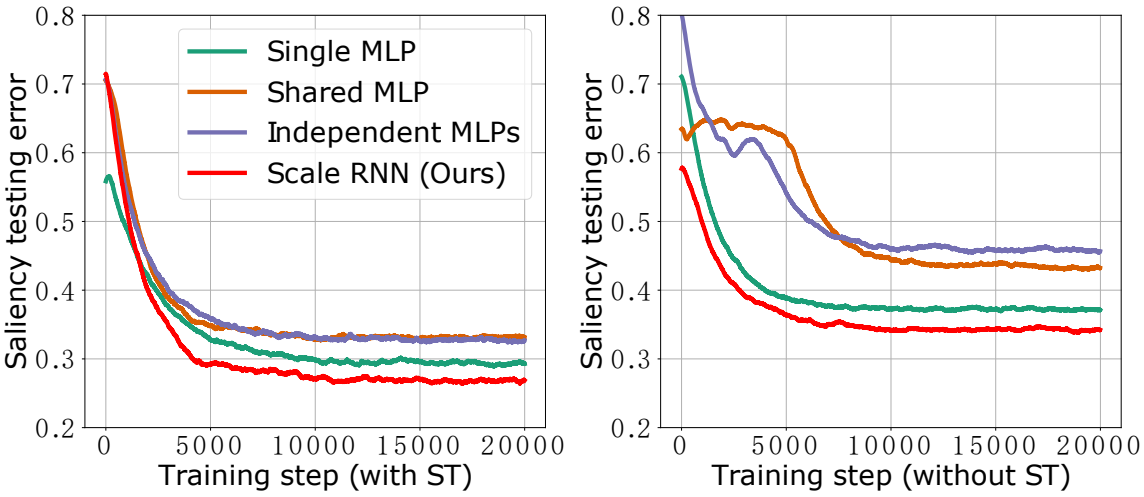


Figure 4.9: **RNN Evaluations**. Our RNN method of learning and integrating multi-scale shape features produces the lowest saliency testing error, among the four feature embedding architectures in Fig. 4.7, both with (*left*) and without the soft-thresholding operator (*right*).

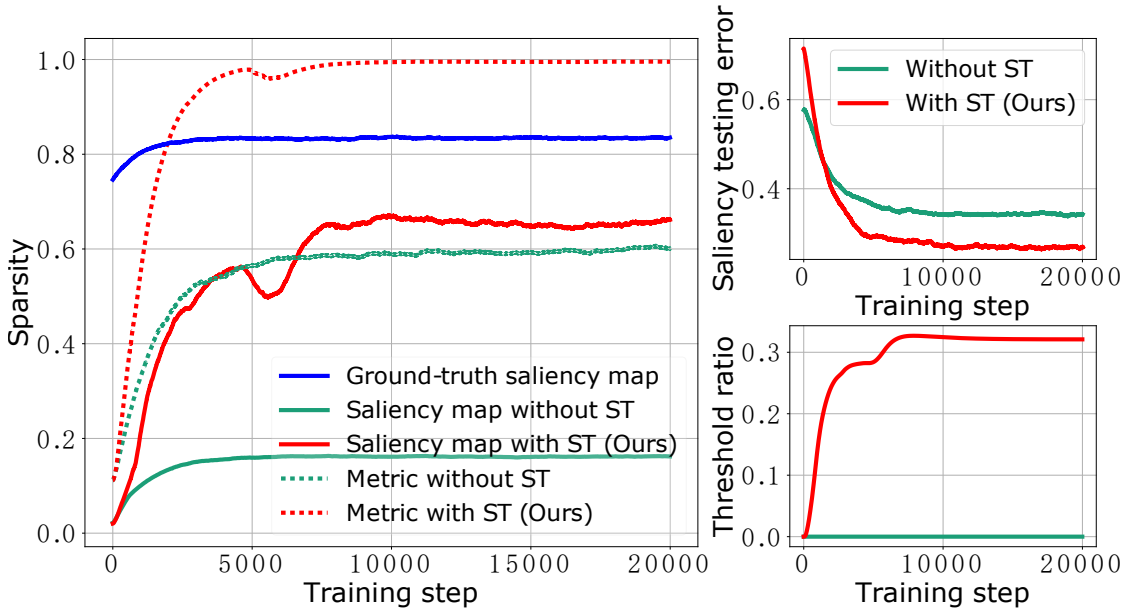


Figure 4.10: **Soft-thresholding Evaluations**. Learning a threshold value (*bottom right*) to adaptively truncate the small elements of a metric to exact zeros considerably improves the sparsity of metric and drive the sparsity of saliency closer to that of the ground-truth (*left*). The resulting saliency testing error is also considerably lower (*top right*).

operator.

Evaluation of Our Soft-thresholding Operator. We then evaluate our soft-thresholding operator by training with and without it, as shown in Fig. 4.10. Without the soft-thresholding operator, although the ground-truth saliency maps gradually sparsifies the learned metric, the predicted saliency maps have relatively lower sparsity and considerable higher saliency testing error. Our operator improves the sparsification of the learned metric significantly by gradually learning a threshold to truncate small values, producing much better saliency maps with higher sparsity and lower testing error.

4.4.3 Mutual Benefits of Saliency and Matching

Here, we validate that jointly learning saliency and matching via our unified metric loss function enables each other to generalise better: while matching improves the accuracy and deformation-invariance of our computed saliency maps, saliency improves the semantic localisation of our learned shape embeddings for more robust matching. We evaluate the saliency fitting loss, saliency consistency loss, and metric consistency loss all together, on the Human category of the Schelling saliency dataset (Chen et al., 2012) and the SCAPE matching dataset (Anguelov et al., 2005) (80% for training and 20% for testing). We perform another evaluation on the Fourleg category of the Schelling saliency dataset and the TOSCA matching dataset (Vlasic et al., 2008).

Quantitative Evaluations. As shown in Fig. 4.11, training with only the saliency fitting loss ($\alpha = 1, \beta = 0, \gamma = 0$) leads to the high saliency and metric consistency errors, indicating that neither the predicted saliency maps nor the metrics are sufficiently invariant to human body shape variations. Adding the saliency consistency loss alone ($\alpha = 1, \beta = .02, \gamma = 0$) improves the deformation-invariance of the predicted saliency maps a lot, but the metric remains sensitive to shape variations because only its principal eigenvector (i.e. saliency map) is regularised to be consistent. This can be seen from the high *other consistency error*, which measures the difference of the metric without its principal eigenvector between two corresponding meshes. Oppositely, adding the metric consistency loss alone ($\alpha = 1, \beta = 0, \gamma = .02$) leads to a more deformation-invariant metric by regularising all eigenvectors together, but is less effective compared with the saliency consistency loss for inducing a deformation-invariant saliency map. In contrast, training with all three losses together ($\alpha = 1, \beta = .02, \gamma = .02$) produces the most deformation-invariant

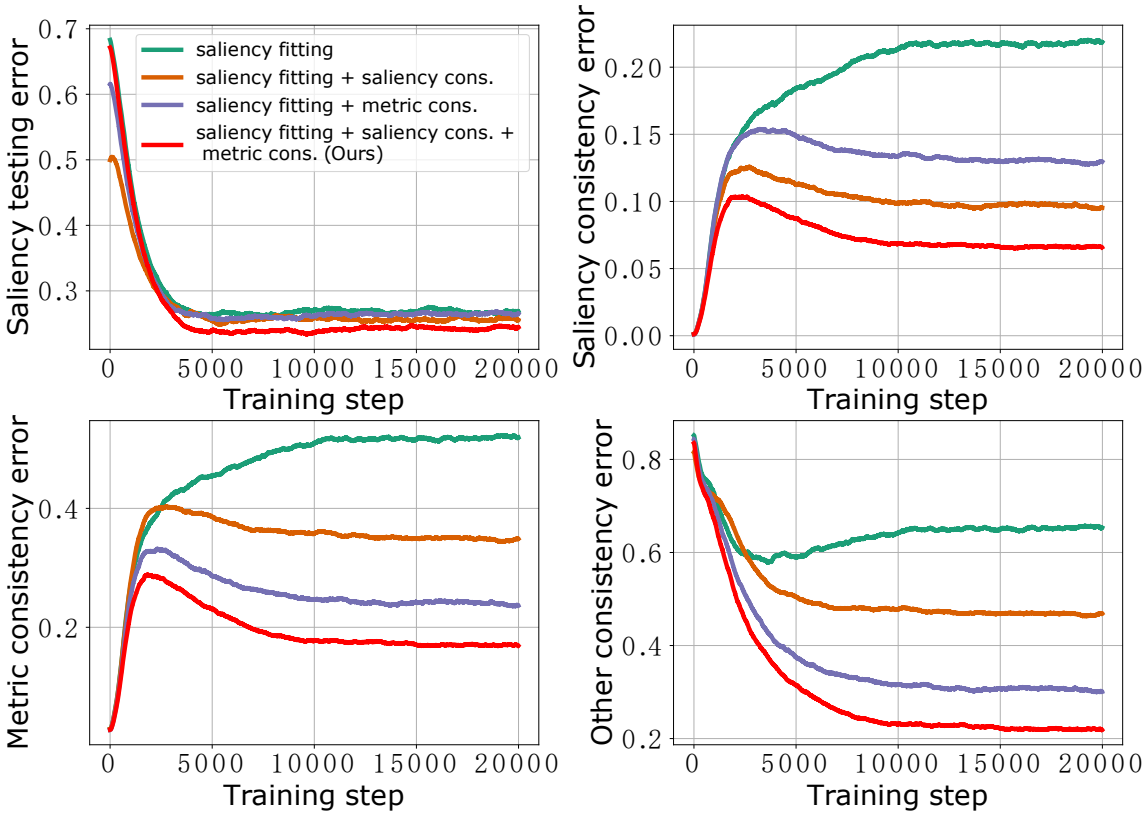


Figure 4.11: **The Quantitative Evaluations of Saliency and Matching**. Learning with the saliency fitting loss, saliency consistency loss, and metric consistency loss together ($\alpha = 1, \beta = .02, \gamma = .02$) produces the lowest errors on all criteria, compared with that when either one of the saliency and metric consistency losses or both are disabled. The *other consistency error* measures the difference of the metric without its principal eigenvector between two corresponding meshes.

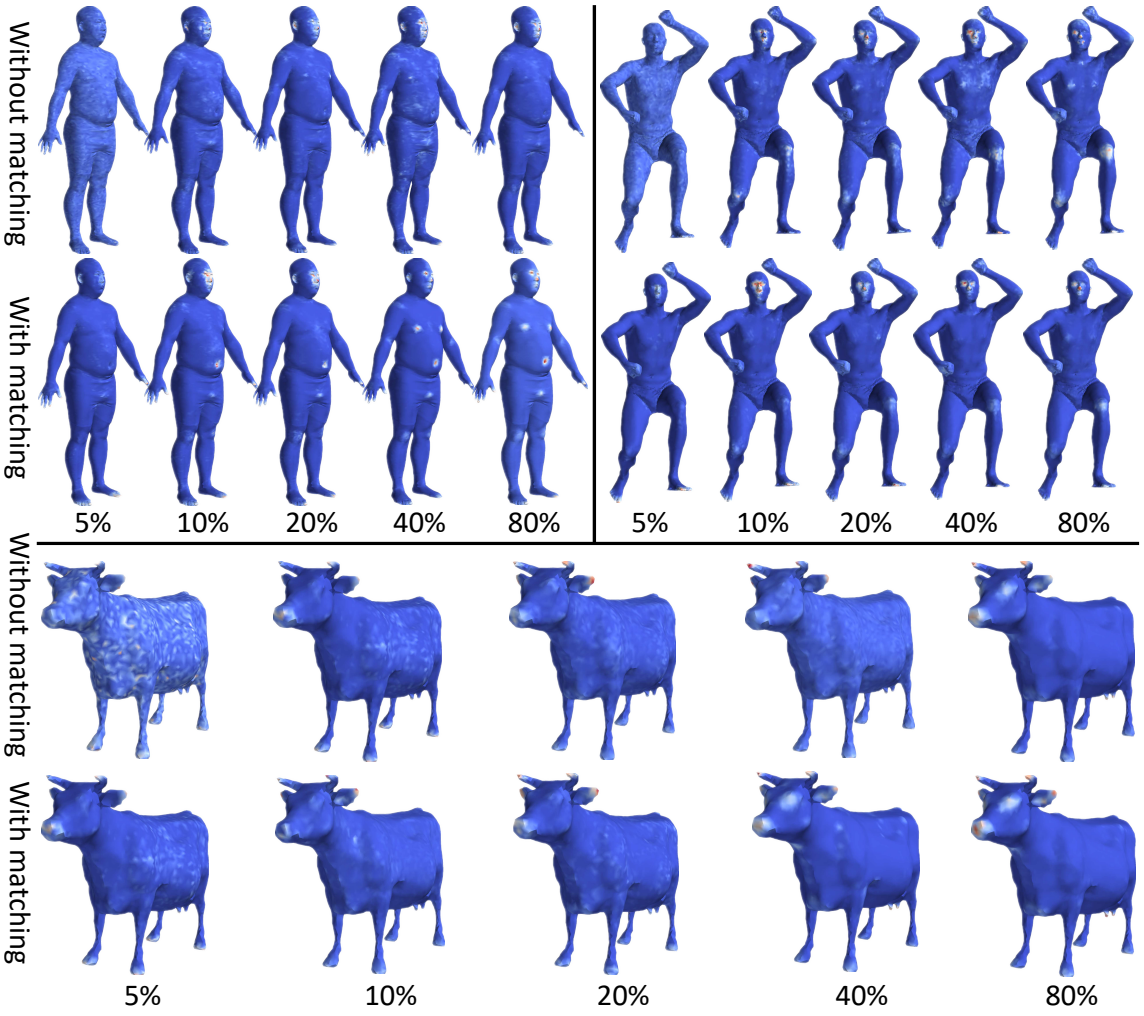


Figure 4.12: **The Benefits from Matching to Saliency**. With matching, our computed saliency maps are less noisy and more sharply highlighted, especially in the extreme case of using one (5%) or two (10%) meshes for saliency training. The visual quality improvement of our saliency maps with matching is still noticeable with more meshes for saliency training.

metrics and saliency maps, while achieving the lowest saliency testing error. The low metric consistency error along with the non-zero eigengaps (Fig. 4.5) ensures that our saliency-induced embeddings are deformation-invariant for shape matching.

Qualitative Evaluations. To compare the predicted saliency maps with and without matching, we train on a 5% (1 mesh), 10% (2 meshes), 20% (4 meshes), 40% (8 meshes), and 80% (16 meshes) samples of the respective dataset with and without the saliency and metric consistency losses. As shown in Fig. 4.12, under the extreme case of a single training mesh, the predicted saliency maps without consistency learning is full of unrecognisable noises. Remarkably, training with shape matchings reduces the noises to a huge extent, allowing the identification of the salient

regions of ears, hands, feet, and facial features. With more training meshes, the predicted saliency maps without consistency learning become less noisy, but they appear quite different between the two testing meshes, which suggests that they are sensitive to the non-rigid shape deformation. In contrast, the maps with consistency learning are much clearer and more consistent, even under the challenging settings of 2 and 4 training meshes. This confirms that overcoming intra-category shape variations via matching is the key to helping saliency detection generalise better, in both small and large sample training scenarios.

We evaluate how saliency can help matching generalise better. We compare our embeddings with the Laplacian spectral embeddings because both of them are eigenvector solutions to the Laplacian embedding problem (4.2a) (4.2b) (4.2c) with salient affinity for the former and cotangent affinity for the later. As shown in Fig. 4.4, our embeddings are perfectly localised on the salient regions of ears, facial features, hands, and feet, while the spectral embeddings are globally supported on the mesh surface. Compared with the existing learned embeddings (Corman et al., 2014; Litman and Bronstein, 2014; Boscaini, Masci, Rodolà, Bronstein and Cremers, 2016; Cosmo et al., 2016; Wei et al., 2016), ours are the first to achieve semantic localisation while guaranteed to be smooth and orthogonal as the spectral embeddings. The semantic localisation property would be difficult to obtain without the use of saliency that agree with human annotations (Chen et al., 2012). As shown in Fig. 4.13, our embeddings discriminate salient points more accurately (left), while maximising the feature invariance among non-salient points since they are less reproducible under intra-category shape deformations (right). In contrast, the spectral embeddings provide an equally rough discrimination accuracy for each point on the shape, irrespective of whether it is salient or not. Our embeddings can therefore be used to prevent erroneous matchings from salient to non-salient points and vice verse, based on the consistency of saliency within a shape category (Chen et al., 2012).

4.4.4 Comparison with Saliency Detection Methods

Here, we compare our method with the highly-cited saliency detection methods, including mesh saliency (MS) (Lee et al., 2005), surface regions of interest (SRI) (Leifman et al., 2012), manifold ranking (MR) (Pingping et al., 2015), spectral irregularity (SI) (Song et al., 2014), tree-based regression (TBR) (Chen et al., 2012), point neural networks (PointNet and PointNet++) (Qi, Su, Mo

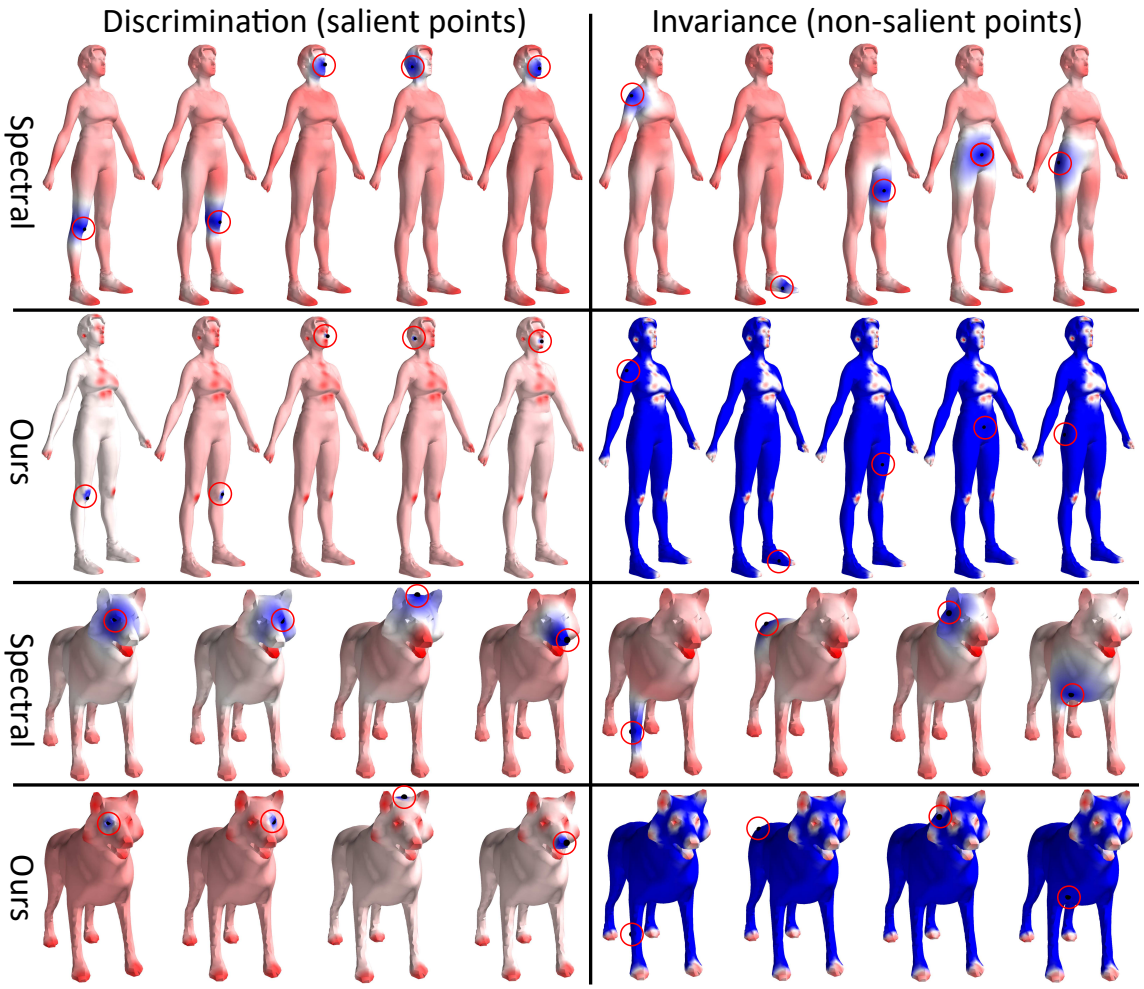


Figure 4.13: **The Benefits from Saliency to Matching**. The red circle on each mesh highlights the reference point, and from there the distances to other points are represented using a blue (small) to red (large) scale. Using salient points as references (*left*), due to the semantic localisation property, our saliency-induced embeddings discriminate these semantically important and thus deformation-stable points much better compared with the isometry-invariant spectral embeddings. Using non-salient points as references (*right*), we achieve maximum invariance for these points that are sensitive to non-isometric deformations.

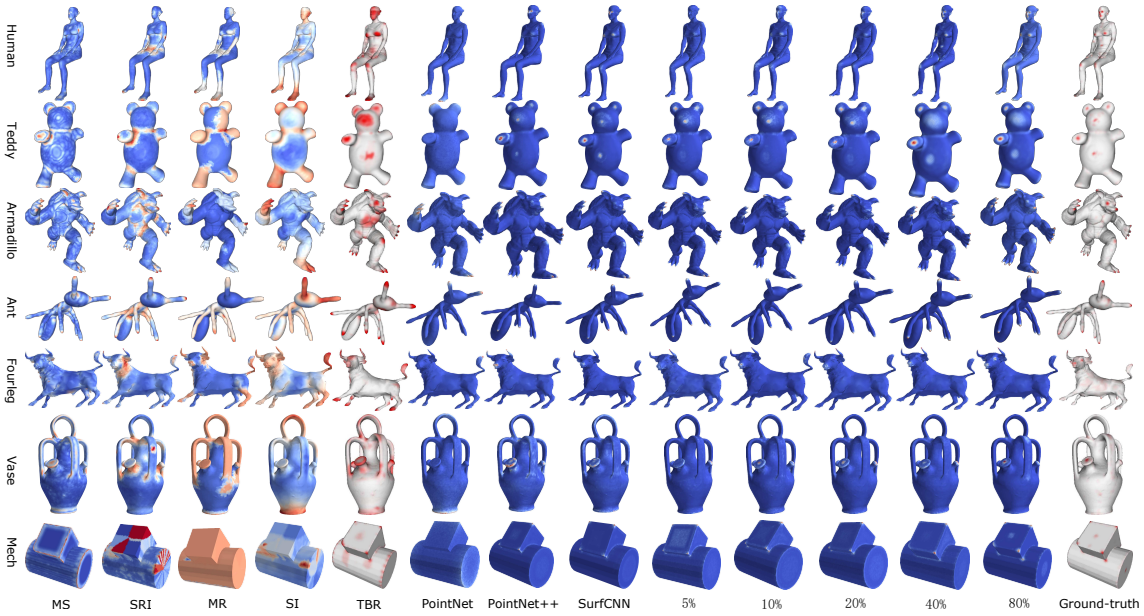


Figure 4.14: **Visual Comparisons for Saliency Detection without Matching**. The image shows the saliency maps generated by MS, SRI, MR, SI, TBR, PointNet, PointNet++, SurfCNN and our method, without the use of matching for saliency detection. Note that while PointNet, PointNet++ and SurfCNN are trained on an 80% sample for all the categories of the Schelling saliency dataset *jointly*, TBR is trained using leaving-one-out for each category *separately* in the original work. Our method is trained on varying fractions of samples for all the categories *jointly* to better visualise progression of generalisation.

and Guibas, 2017*a*; Qi, Yi, Su and Guibas, 2017), and surface CNN (SurfCNN) (Yi et al., 2017). Among them, MS is local contrast-based, SRI, MR, and SI are global rarity-based, and TBR is tree regression-based. Unlike PointNet that works on a raw 3D point cloud, PointNet++ and SurfCNN learn features using the geodesic metric and in the Laplacian spectral domain respectively. We input our raw SH features to PointNet++ and SurfCNN for a fair comparison. Note that our method does not use intrinsic geodesics or Laplacian but may incorporate them in the future.

Saliency Detection without Matching. As MS, SRI, MR, and SI are rule-based and cannot incorporate the intra-category consistency into saliency computation, we first train PointNet, PointNet++, and SurfCNN on an 80% sample (for each category) of the Schelling dataset and our method on a 5%, 10%, 20%, 40%, and 80% sample respectively. We find that our method produces the most accurate saliency maps using 80% training meshes.

Fig. 4.14 shows that MS responds strongly to local geometric variations while SRI, MR, and SI detect more globally distinct regions. As ground-truth saliency maps are spatially localised on surfaces, they must be densely distributed on the frequency dimension due to the well-known un-

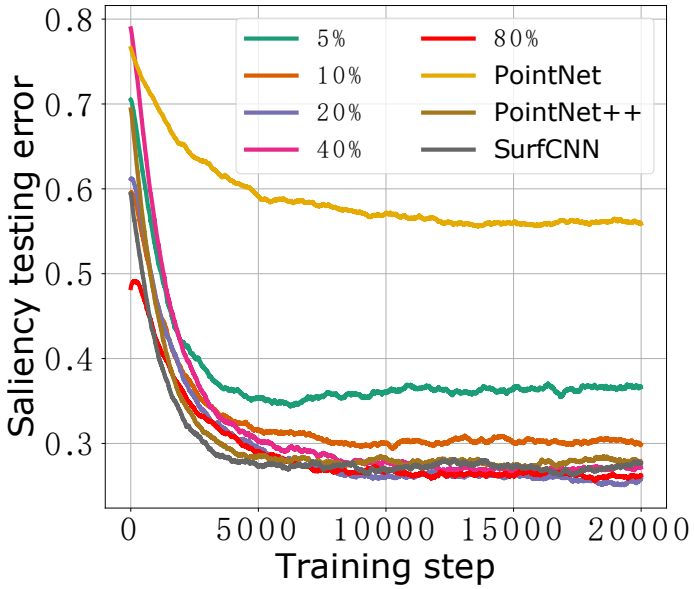


Figure 4.15: **Quantitative Comparisons for Saliency Detection without Matching**. On average, the saliency maps predicted by our method with an 80% training sample are more accurate compared with that by PointNet, PointNet++, and SurfCNN with the same meshes for saliency training.

certainty principle. They are therefore not accurately captured by SI as it involves high-frequency cutoff in spectral computation. TBR produces good saliency maps with leave-one-out training, but fails to well reproduce the sparsity of ground-truth maps (e.g., on the face region of the human body shape). PointNet fails to identify most of ground-truth salient points, which are captured by PointNet++ and SurfCNN to some extent. However, PointNet++ and SurfCNN still misses some important regions such as the mouth and ears of the human body and the eyes of the cow. These regions are accurately captured by our method trained on an 80% sample. Even with as few as 5% or 10% training meshes, our method is shown to detect a succinct set of most important regions such as the facial features and claws of the armadillo.

Fig. 4.15 shows that our method produces more accurate saliency maps than PointNet, PointNet++ and SurfCNN. It is interesting to see that our method achieves equally good quantitative results with a 20% and an 80% sample respectively, but adding more training meshes leads to visually smoother and more accurate saliency maps (Fig. 4.14).

Saliency Detection with Matching. We compare our method with PointNet, PointNet++, and SurfCNN by training with and without the saliency and metric consistency losses on the Human category of the Schelling saliency dataset (Chen et al., 2012) and the SCAPE matching dataset

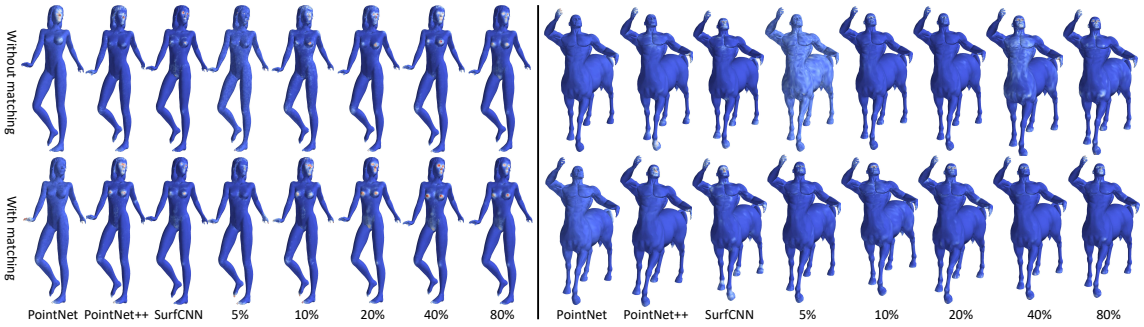


Figure 4.16: **Visual Comparisons for Saliency Detection with Matching**. The image shows the saliency maps generated by PointNet, PointNet++, SurfCNN, and our method, with and without matching. PointNet, PointNet++, and SurfCNN are trained on an 80% sample of the Human category of the Schelling dataset and an 80% sample of the SCAPE dataset, and our method is trained in the same way but with varying fractions of meshes from the Schelling dataset.

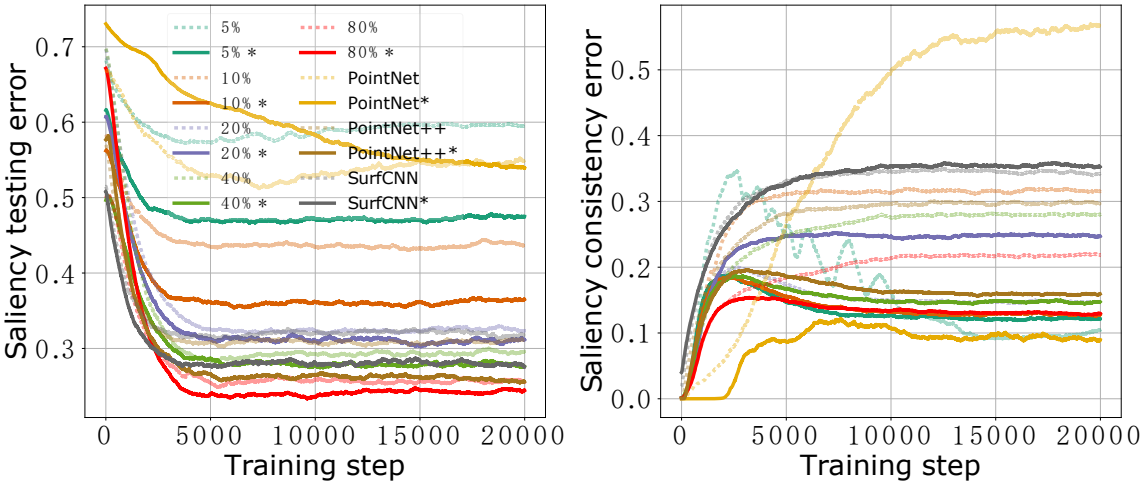


Figure 4.17: **Quantitative Comparisons for Saliency Detection with Matching**. Compared with the saliency maps computed by PointNet, PointNet++, and SurfCNN, ours are more accurate (*left*) and deformation-invariant (*right*). We mark * to indicate the use of matching.

(Anguelov et al., 2005).

Fig. 4.16 shows the predicted saliency maps with and without matching for one testing mesh from the Schelling dataset on the left and another from the TOSCA dataset on the right. Incorporating matching into saliency detection reduces the noises on surfaces to a large extent, especially when there is only 1 training mesh providing no hints about the shape variations of testing meshes. When there are more training meshes, matching is shown to sharpen our detected salient regions such as the facial features of the human body on the left. PointNet fails to detect most of the salient regions, while PointNet++ does not highlight the facial features of the human body clearly. For the centaur shape on the right, PointNet++ and SurfCNN roughly capture the eyes, nose, and

mouth of it with the help of matching. Our method highlights these regions more accurately when matching is used.

Fig. 4.17 shows that enforcing the intra-category consistency of saliency improves the saliency prediction accuracy of all methods except PointNet. The improvement is significant when there are only 1 (5%) or 2 (10%) training meshes but remains noticeable when there are more. Meanwhile, the considerably lower saliency consistency errors indicate that the predicted saliency maps are much more deformation-invariant with matching. Overall, our method achieves the lowest saliency prediction error using 80% of both saliency and matching training meshes.

4.4.5 Comparison with Shape Matching Methods

Here, we compare our method with the highly-cited blended intrinsic maps (BIM) (Kim et al., 2011), semi-definite programming (SDP) (Maron et al., 2016), random forests (RF) (Rodolà et al., 2014), heat kernel CNN (HKCNN) (Boscaini, Masci, Rodolà and Bronstein, 2016), and deep functional maps (DFM) (Litany et al., 2017) for non-rigid shape matching. We group the methods into model-based and learning-based due to their different data requirements - the latter requires one-to-one vertex correspondences for training, while the former does not. We match each of the last 20 testing meshes to the first on the FAUST dataset for performance benchmarking, using the protocol of (Kim et al., 2011). We obtain the correspondences by RF, HKCNN, and DFM for these meshes from the original authors, and run BIM and SDP for the same set of meshes using the released codes.

Saliency for Model-based Matching. We first incorporate our saliency-induced point embeddings (of dimensions 30, Fig. 4.4 bottom) into SDP, in addition to the originally used Laplacian spectral embeddings (of dimensions 30, Fig. 4.4 top), for better handling non-isometric shape deformations. We name our method of SDP with saliency as *SDP-SAL*. Fig. 4.18 shows some predicted correspondence error maps. It can be seen that BIM is inferior to SDP for matching the limbs of human bodies because it has no notion of length on surfaces. SDP produces patches of wrongly matched points due to its sensitivity to surface length-changing (non-isometric) deformations. Our SDP-SAL method reduces the matching errors of SDP at the limbs and chests using the saliency-induced embeddings. Fig. 4.19 shows that our SDP-SAL method achieves higher correspondence accuracy compared with SDP and BIM on the FAUST testing set. The consistent

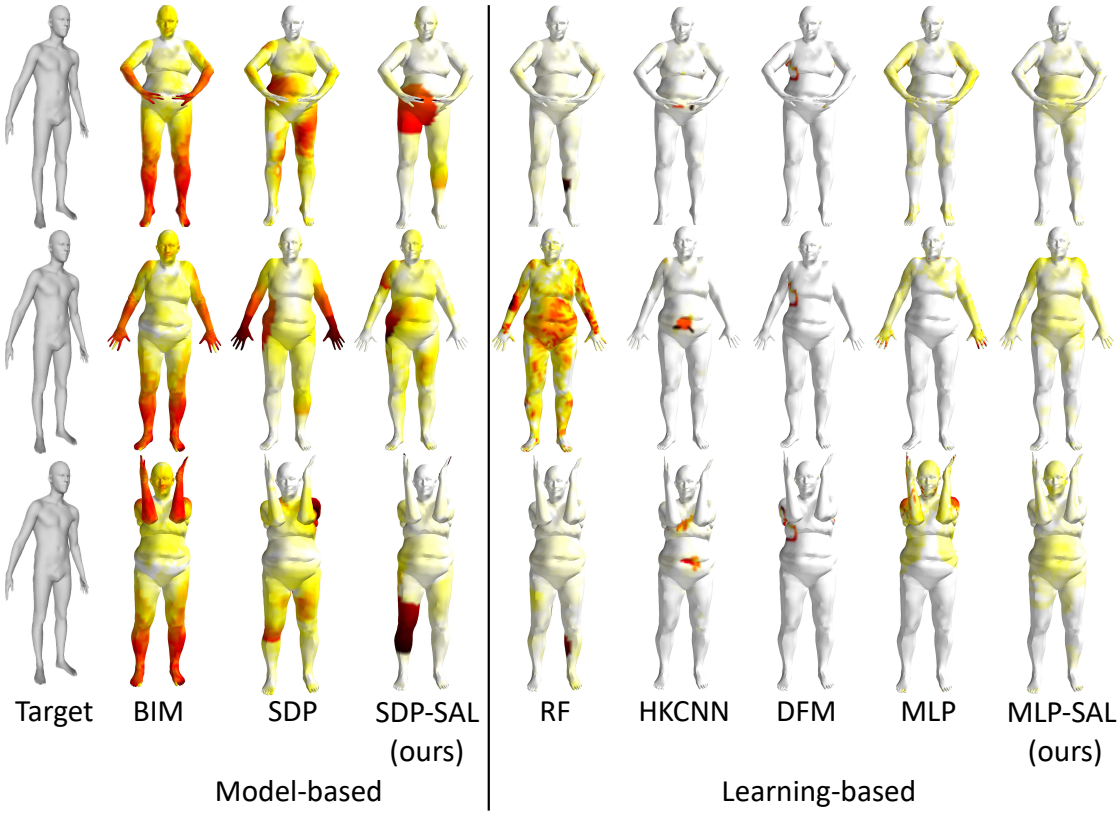


Figure 4.18: **Visual Comparisons for Shape Matching with Saliency**. Visualisation of the predicted correspondence error, i.e. geodesic distances between predicted and ground-truth correspondence points, from three source meshes to a target mesh on the FAUST testing set. Hotter colours indicate larger errors.

improvement from SDP to SDP-SAL indicates that saliency reduces the non-isometric correspondence errors that cannot be handled by the isometry-invariant spectral embeddings.

Saliency for Learning-based Matching. We then incorporate our saliency-induced embeddings into a three-layers plain MLP (of dimensions 256 for each layer) for correspondence prediction using our SH features on the FAUST training set (the first 80 meshes). We name our method of MLP with saliency as *MLP-SAL*. As RF and HKCNN refine the predicted correspondences using the functional maps of (Ovsjanikov et al., 2012) and DFM uses the geodesic smoothing method of (Vestner et al., 2017), we refine our MLP results using the method of (Vestner et al., 2017) for a fair comparison. Our MLP-SAL method exploits both geodesic (as used by DFM) and our saliency-induced embedding distances (Fig. 4.13) for correspondence refinement. Fig. 4.18 shows that our MLP-SAL method reduces the matching errors produced by MLP at the shoulders and hands of human bodies. Fig. 4.19 shows that our MLP-SAL improves on MLP and achieves higher correspondence accuracy compared with RF and HKCNN.

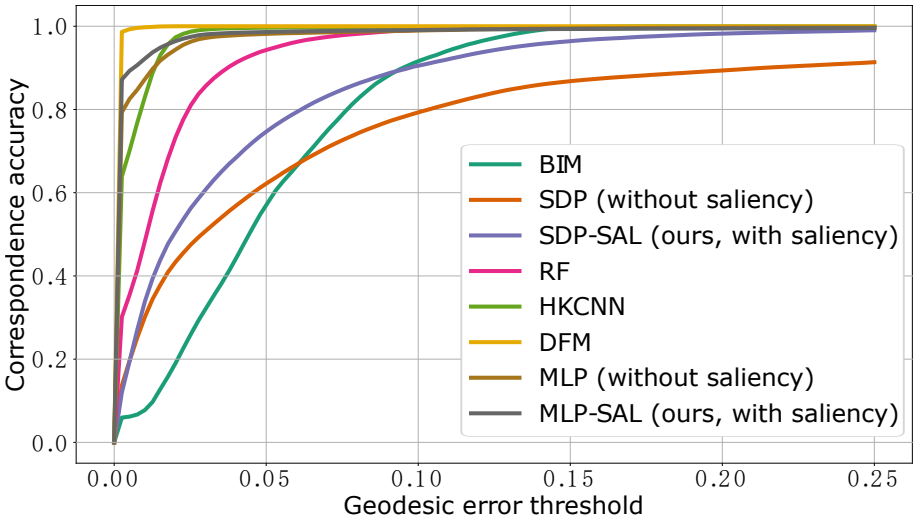


Figure 4.19: **Quantitative Comparisons for Shape Matching with Saliency**. The comparison of the shape matching accuracy obtained by BIM, SDP (without saliency), RF, HKCNN, DFM, MLP (without saliency), as well as by our saliency-enhanced SDP-SAL and MLP-SAL on the FAUST testing set.

More Non-Isometric Matching Results. To demonstrate the use of our saliency-induced embeddings for handling more complex intra-category shape variations, we compute shape matchings for the Fourleg category of the Schelling dataset using the isometry-invariant SDP and our saliency-enhanced SDP-SAL respectively. We extract our embeddings by training with an 80% sample of the Fourleg category and an 80% sample of the animal category of TOSCA shape matching dataset. Fig. 4.20 shows that these animal body shapes have strong non-isometric shape variations, which explains the failure of SDP to find semantically meaningful yet highly non-isometric shape matchings. Our SDP-SAL, in contrast, identifies correct matchings from the limbs of the horse, wolf, and pig to that of the cow. It also considerably reduces the matching errors of SDP at the face and back regions of the wolf and pig. For the even more challenging giraffe-to-cow example, only our SDP-SAL can identify correct matchings for the head region of the giraffe. We note that there are some incorrect discontinuity matchings between the source and the target meshes, which are caused by the used Laplacian embeddings that are sensitive to highly non-isometric shape deformations. In the future work, we plan to address this shortcoming by replacing the Laplacian embeddings with more deformation-invariant embeddings learned from correspondence meshes.

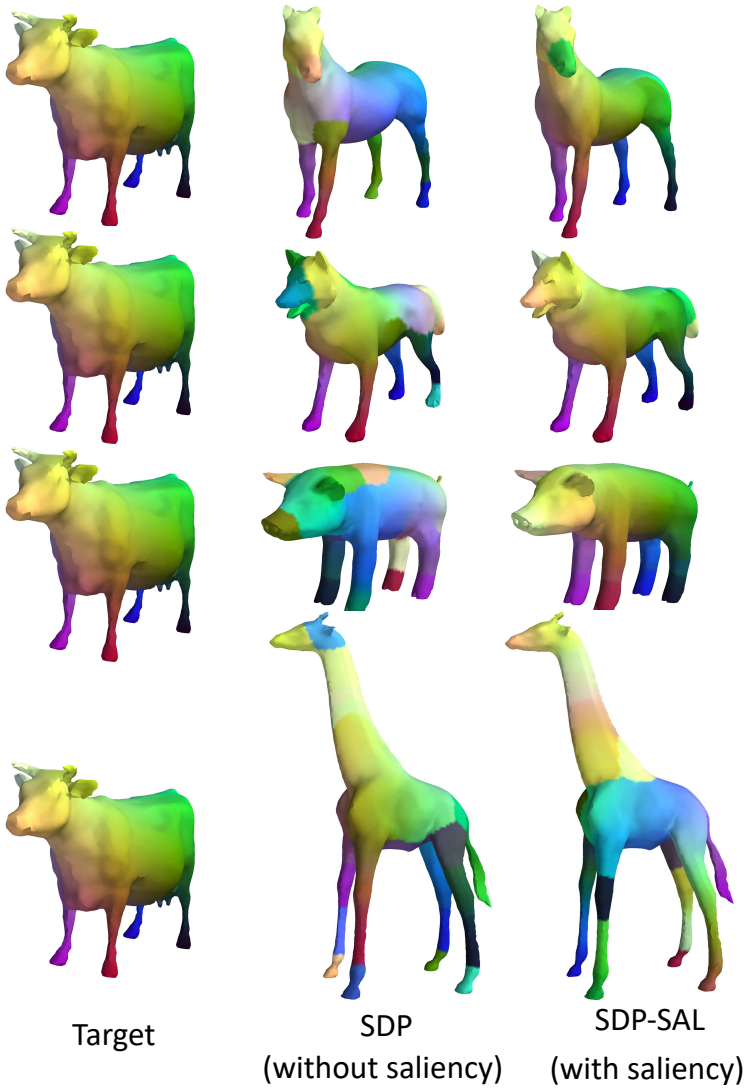


Figure 4.20: **Matching Highly Non-Isometric Shapes with Saliency**. The image shows the shape matchings generated by SDP and our SDP-SAL from four source meshes to a target mesh. These meshes are from the Fourleg category of the Schelling dataset, which is known to exhibit intra-category shape deformations that are far from being isometric.

4.5 Summary

In this chapter, we tackled mesh saliency detection and non-rigid shape matching together for mutual benefits. We proposed a unified metric representation from which the saliency map and the shape embeddings of a mesh can be jointly inferred as the principal eigenvector and the smoothed Laplacian eigenvectors respectively. We also proposed a multi-layer RNN for effectively integrating multi-scale shape features, together with a soft-thresholding operator that adaptively enforces the sparsity of metric representation. We performed metric learning on saliency detection and shape matching datasets at the same time. Results validated that matching improves the accuracy and intra-category consistency of derived saliency maps, especially when the saliency training set is of small size (i.e. with only 1 or 2 meshes). They also showed that saliency improves the matching accuracy of both model-based and learning-based methods, which is more noticeable when large non-isometric deformations are involved.

Chapter 5

Metric-based Facial Shape Beautification

After addressing shape analysis using our metric representation in Chapter 3 and 4, we now proceed to the task of shape synthesis in this chapter. In particular, we propose a novel metric representation for the synthesis problem of 2D facial shape beautification in images. As shown in Fig. 5.1, the theme of our approach is synthesising a beautified metric representation in the facial metric space from the input face dataset and user controls, from which the beautified facial shape can be reconstructed using global optimisation. We detail our approach in the following sections.

5.1 Introduction

Driven by the culture of social media, the need of appearing attractive (Seidman and Miller, 2013) has stimulated a body of research on facial attractiveness analysis (Zhang et al., 2017). Because the geometry of faces has significant influences on the perceived attractiveness (Leyvand et al., 2008), we focus on facial shape beautification in this work (Leyvand et al., 2008), without altering the original makeups and skin textures. This is desirable in many cases where users want to preserve the original skin appearances and only adjust the facial shapes to a minimum extent.

While facial attractiveness is a subjective notion, some studies have shown that regardless of the ages, genders and races of human observers, their preferences towards more attractive faces share

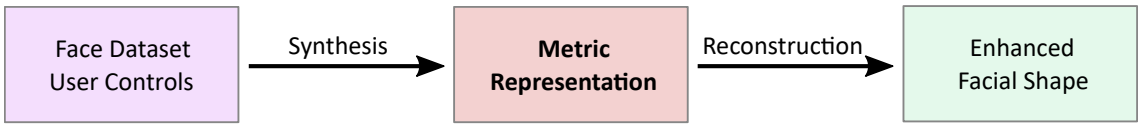


Figure 5.1: **The Overview of Our Unified Metric Framework for Facial Shape Beautification**. Given an input 2D facial image dataset and some user controls, we first synthesise a beautified facial metric representation and then reconstruct the beautified face from the metric.

some objective natures (Cunningham et al., 1995; Slater et al., 1998; Winston et al., 2007). This motivates the use of computer techniques for facial attractiveness analysis and enhancement. There have been studies on the impact of the geometry of faces on attractiveness evaluation (Schmid et al., 2008) and enhancement (Liao et al., 2012). However, their rule-based methods cannot quantify how much a face departs from or conforms to the average of a group of face, which is hypothesised to be attractive (Grammer and Thornhill, 1994). In contrast, data-driven methods that require human-annotated attractiveness scores for face analysis and beautification have been proposed (Eisenthal et al., 2006; Leyvand et al., 2008). Despite their good results on frontal and neutral faces, these methods cannot generalise well to non-frontal, non-neutral faces because the annotation of attractiveness scores is ambiguous when poses and expressions are confounded (Zhang et al., 2017). More recently, deep learning techniques have been applied to the evaluation and enhancement of facial attractiveness (Gan et al., 2014; Li et al., 2015). They suffer from the lack of interpretability of deep neural networks, which prevents users from understanding and controlling the computational process of beautification.

In view of the above challenges, we propose to approach the beautification of non-frontal, non-neutral faces in a geometrically controllable way. Departing from the coordinate-based face representations of (Leyvand et al., 2008; Liao et al., 2012; Chen et al., 2014), we propose to encode the geometry of a face using the orthogonal projection metric onto the subspace of the facial landmarks. Due to the affine-invariance of our metric representation, it frees the face beautification process from the underlying nuisance facial landmarks transformations, such as translation, rotation, and scaling. Furthermore, it fully encodes the pairwise geometric configuration of facial landmarks, which enables users to prescribe different levels of beautification for individual facial parts. Comparing with the non-linear face reconstruction methods in (Leyvand et al., 2008; Liao et al., 2012; Chen et al., 2014), the reconstruction of facial landmarks from our metric representation is a linear projection, which is much more efficient and guaranteed to be globally

optimal.

On top of our metric representation of faces, we propose to identify beautiful face patterns as the local density modes in the metric space of faces. The idea is that these density modes represent the local clusters of facial shapes and exhibit a stronger tendency of symmetry and averageness (Grammer and Thornhill, 1994). We observe that to obtain high-fidelity face beautification results, users typically want to apply minimum changes to the geometry of an input face while keeping the original pose and expression intact. Therefore, we formulate the beautification process of an input face as pulling it towards a local nearby density mode, which can be efficiently found using the mean-shift method (Georgescu et al., 2003). As the method successively averages the face metrics in a local neighbourhood until convergence, it is locality-sensitive averaging for face beautification. The locality-sensitive averaging allows us to adapt the beautification of a particular face to the local vicinity in the metric space of faces, so that the original pose and expression variations can be preserved.

Results show that our method improves facial attractiveness for a wide range of non-frontal poses and non-neutral expressions, without relying on any human-annotated attractiveness scores for training (Leyvand et al., 2008; Chen et al., 2014; Li et al., 2015). 70% of the human subjects we interviewed prefer our beautified results for 100 frontal portraits, while 65% of the subjects prefer the beautified results for 100 general facial images. Results also show that our method preserves user-specified facial parts for face beautification. The whole process takes less than 1 second to finish on a laptop and therefore allows for continuous user interactions. Our publicly available source codes can be downloaded from this link*.

The contributions of this work include:

- We propose to represent the geometry of a face using the orthogonal projection metric onto the subspace of the facial landmarks. This representation is affine-invariant, captures the pairwise geometric configuration among facial landmarks, and allows for efficient face beautification with the user-specified weights of individual facial parts.
- We propose to formulate the beautification process of a face as pulling it towards a local nearby density mode using the mean-shift method. The method is capable of beautifying

*https://drive.google.com/open?id=1NonS5WQedtxejTDh-m_Ym3MZSk21H54p

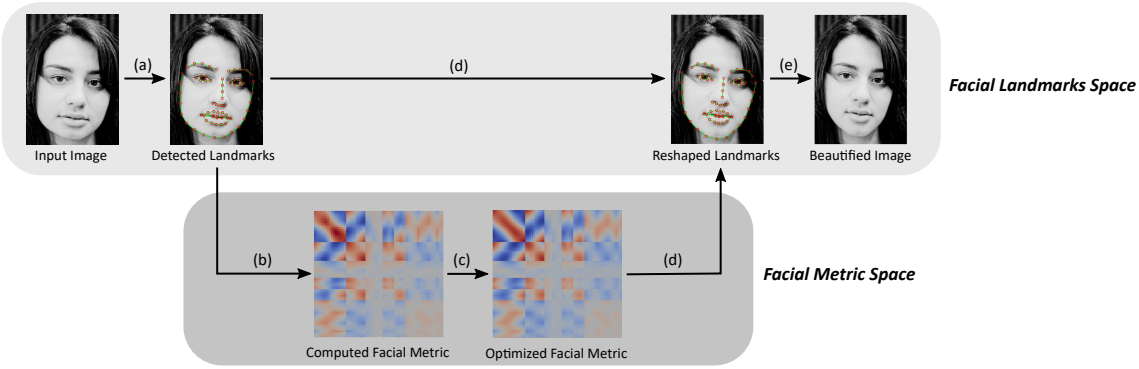


Figure 5.2: **The Overview of Our Metric-based Face Beautification Method**. The computation steps from (*a*) to (*e*) are facial landmarks detection, facial metric computation, facial metric optimisation, facial landmarks reconstruction, and facial image warping respectively.

non-frontal, non-neutral faces by applying minimum changes to input faces via locality-sensitive averaging.

5.2 Our Metric Approach to Face Beautification

Fig. 5.2 provides an overview of our face beautification method. Given an input facial image, we first detect (*a*) a collection of facial landmark points along the jawline and the contours of the eyebrows, eyes, nose, and mouth. We then compute (*b*) a metric to capture the input facial shape, which is the orthogonal projection matrix onto the subspace of the detected landmarks. After that, we perform (*c*) face beautification on the metric by pulling it towards a local nearby density mode in the metric space of faces. We then reshape (*d*) the original landmarks using the optimised metric so that the geometric configuration of the input face can be enhanced. Finally, we warp (*e*) the input image using the original and the reshaped landmarks to produce a beautified image. We describe the steps in the following sections.

5.2.1 Facial Landmarks Detection

Given an input facial image, the first step of our approach is to detect a collection of facial landmarks $\boldsymbol{P} \in \mathbb{R}^{N\times 3}$ for representing the geometry of the input face, where N is the number of landmark points. The coordinates of the i-th landmark is $\boldsymbol{P}_i = (\boldsymbol{P}_{i1}, \boldsymbol{P}_{i2}, 1)$, where $\boldsymbol{P}_{i1}$ and $\boldsymbol{P}_{i2}$ are the x and y coordinates on the image plane respectively. The last component is the homogeneous coordinate 1, which we need to derive our facial metric representation of P in Section 5.2.2.

We adopt the method of (Le et al., 2012) for detecting a number of $N = 68$ facial landmark points along the jawline and the contours of the eyebrows, eyes, nose and mouth. This method builds a local shape model for each individual facial part, which is capable of handling a wide range of facial expression and pose variations.

5.2.2 Facial Metric Representation

After detecting a collection of facial landmarks $\boldsymbol{P}$ from an input image, we aim at deriving a more effective representation of $\boldsymbol{P}$ for facial shape manipulation. Our idea is to represent $\boldsymbol{P}$ as the orthogonal projection metric $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ onto the coordinate subspace of $\boldsymbol{P}$. Traditionally, the normalised coordinates of $\boldsymbol{P}$ and the edge lengths of the Delaunay triangulation have been widely used (Leyvand et al., 2008; Chen et al., 2014). However, these low-level representations cannot be effectively converted back to the coordinate space, making the facial reshaping process non-linear with potentially bad local minimas. In contrast, our metric representation $\boldsymbol{M}$ is naturally invariant to the nuisance affine perturbations of the detected facial landmarks, which geometrically correlates every pair of landmarks in a human-understandable way. In Section 5.2.4, we show that it significantly simplifies the facial reshaping process to linear projection, allowing for flexible user controllability on the beautification process

Formulation. To solve for the metric representation $\boldsymbol{M}$ that geometrically correlates every pair of facial landmarks, we consider representing each landmark $\boldsymbol{P}_i$ as the linear combination of all the landmarks. Because the number of landmarks N is normally greater than the coordinate dimension (i.e. 3), we further minimise the L2-norm of the combination coefficients so that the representation can be uniquely determined. This leads to the following constrained minimisation problem for each landmark:

$$\boldsymbol{x}^* = \arg\min \|\boldsymbol{x}\|_2^2, \text{ subject to } \boldsymbol{P}_i = \sum_{k=1}^{N} \boldsymbol{x}_k \boldsymbol{P}_k \tag{5.1}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is a vector of linear combination coefficients for each landmark and $\|\boldsymbol{x}\|_2^2 = \sum_{k=1}^{N} \boldsymbol{x}_k^2$ is the L2-norm of $\boldsymbol{x}$ to be minimised. Solving (5.1) for each landmark individually

gives us the metric representation $\boldsymbol{M}$:

$$\overbrace{\boldsymbol{M} = \boldsymbol{P}(\boldsymbol{P}^T\boldsymbol{P})^{-1}\boldsymbol{P}^T}^{\textit{Facial Reshaping Metric}} \tag{5.2}$$

where T is the matrix transpose and each row of $\boldsymbol{M}$ is the optimal solution to (5.1) for the corresponding landmark. We can verify from (5.2) that the representation is symmetric ($\boldsymbol{M} = \boldsymbol{M}^T$), idempotent ($\boldsymbol{M} = \boldsymbol{M}^2$), and exact ($\boldsymbol{P} = \boldsymbol{M}\boldsymbol{P}$), confirming that it is the orthogonal projection metric onto the subspace of the facial landmarks. This construction fundamentally changes the formulation of facial shape manipulation from the traditional non-linear coordinates optimisation to our linear projection.

Affine-Invariance. Here, we verify that our metric representation $\boldsymbol{M}$ is guaranteed to be affine-invariant, which allows for efficient facial shape analysis and manipulation without nuisance facial landmark coordinate transformations involved in the process. For each landmark $\boldsymbol{P}_i$, it can be seen from (5.1) that the constraint still holds when we apply any linear transformation $\boldsymbol{L} \in \mathbb{R}^{3\times3}$ to all the landmarks, $\boldsymbol{L}\boldsymbol{P}_i = \sum_{k=1}^{n} \boldsymbol{x}_k(\boldsymbol{L}\boldsymbol{P}_k)$. This shows that the optimal combination coefficients (i.e. rows of $\boldsymbol{M}$) remain the same under any linear (rotation, uniform scaling, and shearing) perturbation in real-world facial images. By applying any homogeneous translation $\boldsymbol{T} = (x, y, 1) \in \mathbb{R}^3$ to all the landmarks, we can further obtain $\boldsymbol{P}_i + \boldsymbol{T} = \sum_{k=1}^{N} \boldsymbol{x}_k(\boldsymbol{P}_k + \boldsymbol{T}) = \sum_{k=1}^{N} \boldsymbol{x}_k\boldsymbol{P}_k + (\sum_{k=1}^{N} \boldsymbol{x}_k)\boldsymbol{T}$. Because the last coordinate components of all the landmarks are 1, we have $\sum_{k=1}^{N} \boldsymbol{x}_k = 1$ from the constraint in (5.1). Therefore, the equation $P_i + T = \sum_{k=1}^{N} \boldsymbol{x}_k(\boldsymbol{P}_k + \boldsymbol{T})$ holds, which confirms that the combination coefficients are also invariant to any translation of the facial landmarks.

Geometric Significance. On top of the affine-invariance, our metric representation $\boldsymbol{M}$ also has clear geometric significance that is not provided by deep learning methods (Gan et al., 2014; Li et al., 2015). This allows it to fully capture the geometric configurations for every pair of facial landmarks, which enables part-based user control in face beautification. We reveal this by finding the analytic form of each element of $\boldsymbol{M}$ as follows:

$$\boldsymbol{M}_{ij} = \frac{(\boldsymbol{P}_i - \overline{\boldsymbol{P}})^T\boldsymbol{S}^{-1}(\boldsymbol{P}_j - \overline{\boldsymbol{P}})}{N} + \frac{1}{N} \tag{5.3}$$

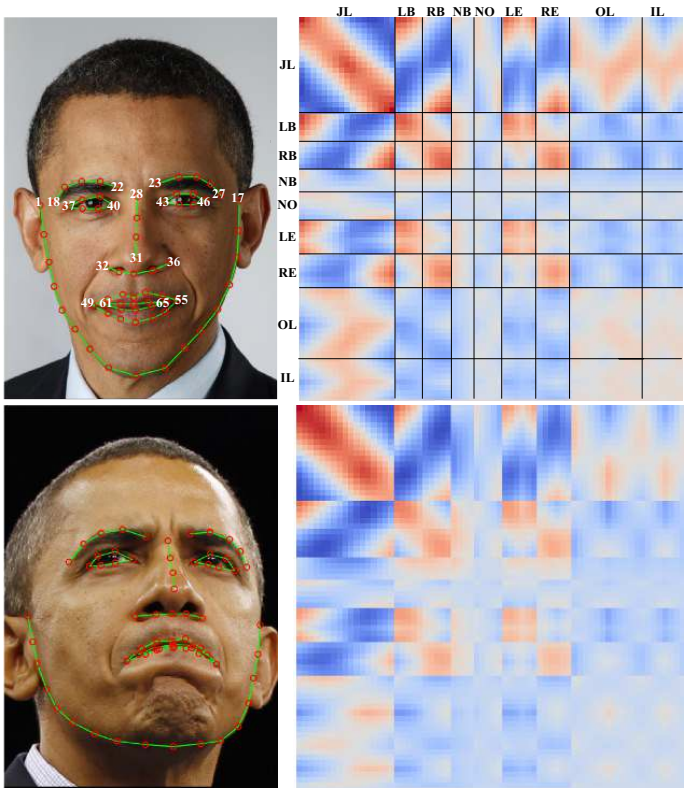


Figure 5.3: **The Visualisation of Our Metric Representation of Facial Landmarks**. *Left*: two faces with different expressions and poses, with the corresponding facial landmarks rendered on top of the faces; the white numbers on the top left show the indices of the 68 facial landmarks. *Right*: the rendered images of the two corresponding metric representations, with blue and red colours representing low and high values respectively; the black lines separate different facial parts for clearer visualisation. These parts include the jawline (**JL**), left eyebrow (**LB**), right eyebrow (**RB**), nose bridge (**NB**), nostril (**NO**), left eye (**LE**), right eye (**RE**), outer lip (**OL**), and inner lip (**IL**). The pairwise geometric configurations represented by our metric enables part-based user control in face beautification.

where $\overline{\boldsymbol{P}} = \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{P}_k$ is the centroid of the face and $S = \frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{P}_k - \overline{\boldsymbol{P}})(\boldsymbol{P}_k - \overline{\boldsymbol{P}})^T$ is the covariance matrix of the landmark coordinates. Now, it becomes clear that the diagonal elements of $\boldsymbol{M}$ depend on the normalised squared Euclidean distance from each landmark to the face centroid, while the non-diagonal elements depend on the dot product between the pair of normalised vectors from the two corresponding landmarks to the centroid.

Facial Metric Visualisations. To demonstrate the geometric significance of our metric representation, we visualise the metrics of two faces with different expressions and poses in Fig. 5.3. It can be seen that each facial part has its a distinct pattern within the matrix. These patterns come from the configurations of the facial parts relative to each other, which vary according to the facial shape. As the distances from the landmarks to the face centres are encoded in the diagonals of the metrics, the elements corresponding to the outermost contours (i.e. the jawline and eyebrows) have relatively larger values. The inner facial parts such as the lips and nose have relatively smaller values. The main difference between the two metrics are on the rows and columns corresponding to the jawline and lips, due to the more significant changes caused by expression and pose. It is also possible to infer the symmetry information of facial parts from the visualised metrics. Taking the first one for example, the submatrix occupied by the jawline and eyebrows is nearly symmetric about its main diagonal, signifying the vertical symmetries of the two facial parts. In comparison, the symmetry patterns of the second face have been considerably weakened by the changing expression and pose.

5.2.3 Facial Metric Optimisation

After computing the metric $\boldsymbol{M}$ from an input set of facial landmarks $\boldsymbol{P}$, the next step of our approach is to optimise $\boldsymbol{M}$ into $\boldsymbol{M}^*$ with improved facial attractiveness. The main idea of our method is to iteratively pull $\boldsymbol{M}$ towards a local nearby density mode in the metric space of faces. This allows us to preserve the original pose and expression during beautification, while gradually improving the attractiveness of the represented face. After obtaining the optimised metric, we apply it to the input facial landmarks to obtain the reshaped landmarks. The reconstructed facial image is generated by smoothly deforming the input image so that the input facial landmarks are aligned to the reshaped ones.

We require a dataset of un-annotated facial metrics $\mathcal{M} = \{\boldsymbol{M}^i\}_{i=1}^{m}$ for training, where m is the

number of faces and each metric $M^i \in \mathbb{R}^{N\times N}$ is computed from the corresponding face. While the methods of (Leyvand et al., 2008; Chen et al., 2014; Li et al., 2015) focus on the beautification of frontal and neutral faces, our method can enhance the shapes of faces with non-frontal poses and non-neutral expressions, without requiring human-annotated attractiveness scores for training.

Feature Transformation. To preserve the major features of faces in beautification, we propose to transform the original metric representation using the Principal Component Analysis (PCA) for dimension reduction and selection. By preserving 95% of the dataset variance, we obtain a low-dimensional close approximation of an input metric $\boldsymbol{M}$ as follows:

$$\boldsymbol{M} = \overline{\boldsymbol{M}} + \sum_{c=1}^{C} \lambda_c \boldsymbol{\Gamma}_c \tag{5.4}$$

where $\overline{\boldsymbol{M}} \in \mathbb{R}^{N\times N}$ is the mean, $\{\boldsymbol{\Gamma}_c\}_{c=1}^{C}$ are the principal components in the descending order of the associated variance, and $\{\lambda_c\}_{c=1}^{C}$ are the projection coefficients of $\boldsymbol{M}$ onto these components. Therefore, the metric can be approximated using the mean metric and a set of mutually uncorrelated details. These details are sorted in the descending order of the corresponding principal eigenvalues (i.e. data variances).

As the leading coefficients explain more rapidly changing expressions and poses in the dataset, we choose to use the non-leading coefficients for facial shape optimisation. Empirically, we exclude the first two coefficients and subject the remaining $\{\lambda_c\}_{c=3}^{C}$ to optimisation. We reconstruct the optimised metric as $\boldsymbol{M}^* = \overline{\boldsymbol{M}} + \sum_{c=1}^{2} \lambda_c \boldsymbol{\Gamma}_c + \sum_{c=3}^{C} \lambda_c^* \boldsymbol{\Gamma}_c$, where $\{\lambda_c^*\}_{c=3}^{C}$ are the coefficients after face optimisation. This is visualised in Fig. 5.4. In the following, we use $\boldsymbol{M}$ to abbreviate the coefficient representations $\{\lambda_c\}_{c=3}^{C}$ for explaining our face beautification method.

We show an example of face interpolation in Fig. 5.5, where we apply the mean of the facial metrics of two source faces to an input face with and without feature transformation. We evaluate this task because the averaging operation is the building block of our unsupervised face beautification method. It can be seen that while the two source faces have different expressions and poses, they share the consistent intrinsic facial geometric configuration. However, when their shapes are naively interpolated and transferred to a target face, some extrinsic factors creep in and lead to undesirable face distortions. By excluding the leading PCA coefficients, it can be seen that the

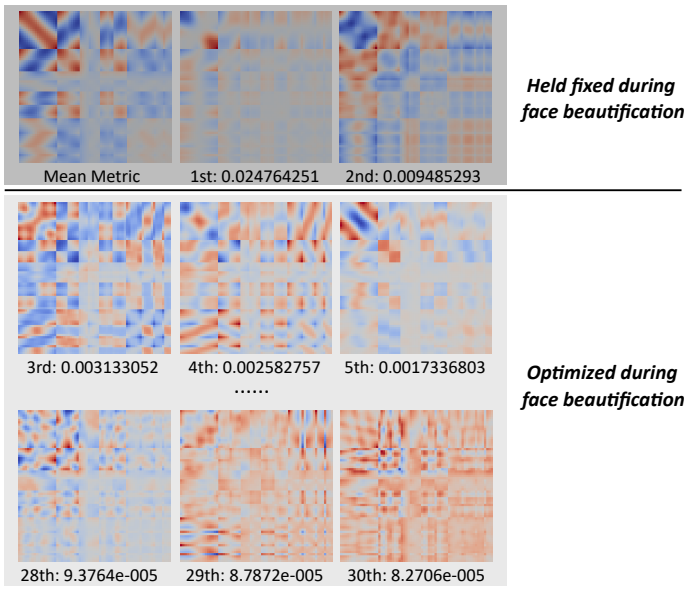


Figure 5.4: **The Visualisation of Our PCA Feature Transformation**. The images show the mean facial metric and the discovered PCA components as sorted by their associated eigenvalues (displayed on the bottom of each small image). We fix the PCA coefficients corresponding to the first and the second components during face beautification, because they encode large-scale pose and expression variations and modifying them leads to noticeable facial image distortions. We only optimise the remaining PCA coefficients that mostly encode intrinsic facial shape features.

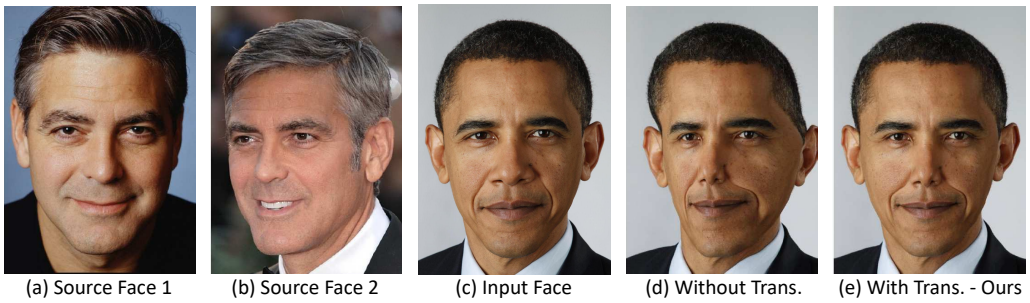
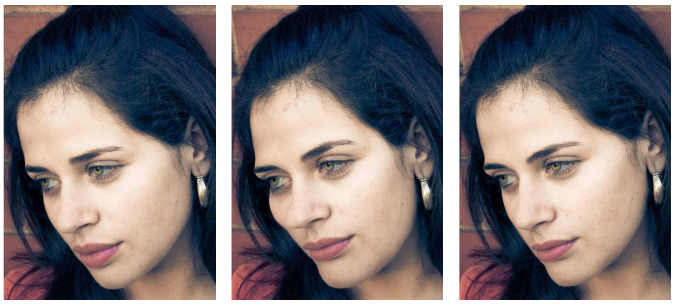


Figure 5.5: **The Effect of Feature Transformation on Facial Shape Interpolation**. (*a*): a frontal and neutral source face. (*b*): another source face with pose and expression. (*c*): an input face. (*d*): the interpolated face generated by applying the mean of the two source facial metrics to the input face without feature transformation. (*e*): the interpolated face generated using our feature transformation method.



(a) Input Face (b) Without Trans. (c) With Trans. - Ours

Figure 5.6: **The Effect of Feature Transformation on Face Beautification**. (*a*): an input face. (*b*): the beautified face without using feature transformation. (*c*): the beautified face with our feature transformation method.

transferred face has no noticeable distortions. Furthermore, we show an example of face beautification with and without feature transformation in Fig. 5.6. It can be seen that the beautified image with feature transformation is much more realistic compared with that without feature transformation. The output images in Fig. 5.5 and 5.6 are generated using the facial image warping method in Section 5.2.5.

Locality-sensitive Face Beautification. Motivated by the averageness property of facial attractiveness (Grammer and Thornhill, 1994; Schmid et al., 2008; Zhang et al., 2011), we propose to optimise an input facial metric $\boldsymbol{M}$ by iteratively averaging its local nearby samples in the metric space of faces. Our idea is to regard faces at the local cluster centres (i.e. local average faces) as the locally most attractive, which to our knowledge is first proposed in this work. Different from the methods of (Grammer and Thornhill, 1994; Schmid et al., 2008; Zhang et al., 2011), we ensure the locality of the averaging operation so that the averaged faces have improved attractiveness while remaining similar to the input. According to the mean-shift formulation of (Comaniciu and Meer, 2002), this iterative locality-sensitive averaging operation is guaranteed to converge to a local nearby density mode in the metric space, under mild assumptions of the kernel function used for measuring the distance between any pair of facial metrics.

We formulate the idea of iterative local averaging and obtain our face beautification formula at one scale as follows:

$$\overbrace{\Psi(\boldsymbol{M}) = \frac{\sum_{i=1}^{m}[h_i^{-C}e^{-||\boldsymbol{M}^i-\boldsymbol{M}||^2/2h_i^2}]\boldsymbol{M}^i}{\sum_{i=1}^{m}[h_i^{-C}e^{-||\boldsymbol{M}^i-\boldsymbol{M}||^2/2h_i^2}]}}^{\textit{Locality-sensitive Face Averaging}} \tag{5.5}$$

where $\boldsymbol{M}$ is an input facial metric using our PCA feature representation, each $\boldsymbol{M}^i$ is a training

sample from the face dataset $\mathcal{M}$, m is the number of training samples in $\mathcal{M}$, and C is the dimension of the PCA representation. It can be seen from (5.5) that the distance between $\boldsymbol{M}$ and $\boldsymbol{M}^i$ is measured as the Gaussian kernel function $e^{-||\boldsymbol{M}^i-\boldsymbol{M}||^2/2h_i^2}$, which assigns higher weights to geometrically more similar faces and lower weights to faces that are farther away from the input. Compared with the equal or linear weighting kernels, the exponential locality of the Gaussian kernel function allows for better preservation of the input pose and expression during beautification.

To account for varying sample densities, we associate an adaptive scale h_i with each face in the dataset:

$$h_i = ||\boldsymbol{M}^i - \boldsymbol{M}^{\gamma m}||^2 \tag{5.6}$$

where $\gamma \in (0, 1)$ and $\boldsymbol{M}^{\gamma m}$ is the γm-nearest neighbour to $\boldsymbol{M}^i$ in the dataset. When γ is set to a small value, only very close samples can contribute to the averaging and thus only small-scale facial features will be modified. When γ is set to a larger value, more distant samples will be considered and larger-scale facial features will be enhanced. Empirically, we choose the set of scales $\gamma \in \{0.005, 0.01, 0.02, 0.04, 0.08\}$ because they are able to cover both local nearby and more distant training samples for facial shape optimisation. For each γ, we compute the corresponding adaptive scale h_i for each training sample $\boldsymbol{M}^i$ and iteratively evaluate the locality-sensitive averaging operation in (5.5) until convergence. The final optimised metric $\boldsymbol{M}^*$ is computed as the mean of the convergent solutions of all scales.

5.2.4 Facial Landmarks Reshaping

With the original facial landmarks $\boldsymbol{P}$ and the optimised facial metric $\boldsymbol{M}^*$ computed by our locality-sensitive averaging method, the next step of our approach is to find a set of new landmarks $\boldsymbol{P}^*$ that is consistent with the enhanced facial shape represented by $\boldsymbol{M}^*$. We show that our metric representation reduces the process to a linear projection, which is significantly less costly compared with the non-linear optimisation method of (Leyvand et al., 2008; Chen et al., 2014). Besides the global optimality of our method, it allows users to flexibly control the beautification weight of each individual facial part, thereby achieving user-satisfied beautification results.

Formulation. To ensure the quality of face beautification, we minimise the distance between the original and the new landmarks while enforcing the consistency of the new landmarks with the

(a) Source Face (b) Input Face (c) Beautified Face

Figure 5.7: **The Example of Face Beautification via Shape Transfer**. (*a*): a source face. (*b*): an input face to be beautified. (*c*): the beautified face with the shape coming from the example. The shape transfer is efficiently done by applying the metric of the source face to the input as linear projection.

optimised metric:

$$\boldsymbol{P}^* = \arg\min \|\boldsymbol{X} - \boldsymbol{P}\|_F^2, \text{ subject to } \boldsymbol{X} = \boldsymbol{M}^*\boldsymbol{X} \tag{5.7}$$

where $\boldsymbol{X} \in \mathbb{R}^{N\times 3}$ is a set of new landmarks to be found and $\|\boldsymbol{X}-\boldsymbol{P}\|_F^2 = \sum_{i=1}^{N}\sum_{j=1}^{2}(\boldsymbol{X}_{ij} - \boldsymbol{P}_{ij})^2$ measures the deviation of the two sets of landmarks. The constraint enforces that the new landmarks should be within the subspace corresponding to the metric $\boldsymbol{M}^*$, thereby reconstructing the encoded pairwise geometric configurations of the facial landmarks. The optimal solution to (5.7) is the linear projection of the original facial landmarks onto the new optimised metric:

$$\overbrace{\boldsymbol{P}^* = \boldsymbol{M}^*\boldsymbol{P}}^{\textit{Facial Reshaping}} \tag{5.8}$$

Essentially, $\boldsymbol{M}^*$ acts as the facial reshaping metric that enhances an input face via linear projection, which is significantly more efficient than the non-linear optimisation process of (Leyvand et al., 2008; Chen et al., 2014).

Fig. 5.7 shows an example of face beautification via shape transfer. It can be seen that the modified face appears to resemble the source face as indicated by the thinner jawline, longer eyebrows, bigger eyes and mouth. This is efficiently done by linearly projecting the input facial landmarks using the metric of the source face.

User Controllability. Now, we consider the problem of user controllability in face beautification. Sometimes, users may want to prescribe different levels of beautification for individual facial

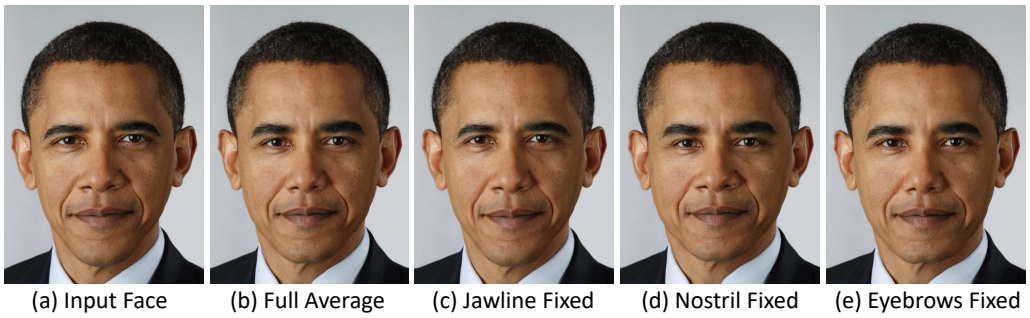


Figure 5.8: **The Example of User-controlled Face Averaging**. (*a*): an input face. (*b*): a full average face created by applying the mean facial metric to the input. (*c*): the average face with the original jawline shape by setting the corresponding beautification weight to 0. (*d*): with the original nostril shape. (*e*): with the original left and right eyebrow shapes.

features, such as preserving the shapes of the eyebrows more while enhancing the shapes of the jawline more. This can be naturally done using our metric representation. We denote a vector of beautification weights as $\mathbf{w} \in \mathbb{R}^N$, where $0 \leq \mathbf{w}_i \leq 1$ is a user-specified value controlling how much the facial shape related to the i-th landmark can be altered. We group the landmarks into the *jawline*, *left eyebrow*, *right eyebrow*, *nose bridge*, *nostril*, *left eye*, *right eye*, *outer lip*, and *inner lip*, as visualised in Fig. 5.3. The landmark weights within each group are equal. As a result, the beautification weight matrix $W = \mathbf{w}\mathbf{w}^T \in \mathbb{R}^{N \times N}$ indicates the weight configuration of the whole face: W_{ij} is larger if the relative geometric configuration of the landmarks i and j should be enhanced more, and W_{ij} is smaller if the configuration needs to be preserved more. We compute the reshaped facial landmarks as follows:

$$\overbrace{\boldsymbol{P}^* = [(1-W)\boldsymbol{M} + W\boldsymbol{M}^*]\boldsymbol{P}}^{\textit{User-controlled Facial Reshaping}} \tag{5.9}$$

where $\boldsymbol{M}$ and $\boldsymbol{M}^*$ are the original and the optimised facial metrics respectively. If $\mathbf{w}$ is a vector of 0, no beautification is applied and the original face is recovered. If $\mathbf{w}$ is a vector of 1, the new face is generated without preserving any features of the original. By adjusting the weights of the facial parts, users can enhance input faces while preserving certain desired features.

Fig. 5.8 shows an example of applying the mean facial metric to reshape an input face, while keeping some of the original facial parts fixed during reshaping. The mean metric is computed on the training set of (Le et al., 2012) and the created full average face appears to have more round and symmetric facial features than the input. By setting the beautification weights of the facial

landmarks on the jawline to 0, the partial average face looks more similar to the input than the full average, as the jawline has a global influence on the shape of the whole face. By fixing the nostril or the eyebrows instead, the results are more similar to the full average face, but the shapes of the nostril and eyebrows still resemble the originals. Our user-controlled facial reshaping method produces satisfactory customised results in all cases.

5.2.5 Facial Image Warping

Finally, given an input facial image and the detected facial landmarks $\boldsymbol{P}$, the final step of our approach is to produce a beautified image using the optimised facial landmarks $\boldsymbol{P}^*$. Our goal is to geometrically deform the input image so that the original facial landmarks can be matched to the optimised ones, while ensuring that the deformed image remains realistic and high-quality. To this end, we adopt the Moving Least Squares method of (Schaefer et al., 2006) for fitting a rigid transformation on each image location. To more efficiently process high-resolution facial images that are popular nowadays, we modify the original method by sampling a low-resolution grid (10% of the image resolution) on the input image and deforming the grid using the original and the optimised landmarks. Afterwards, for each quad on the deformed grid, we compute a backward perspective transformation that maps the quad to the corresponding one on the original grid. Finally, we use these backward transformations to fill the output image using the corresponding pixels from the input image. Our modified method is capable of producing a high-quality, high-resolution beautified image within a second. We use it to generate all of the modified face images in this work.

5.3 Results

5.3.1 Full-face Beautification Results

We show our full-face beautification results in Fig. 5.9. We generate these results using the popular Helen facial image dataset (Le et al., 2012), which consists of $11,147$ high-resolution facial images for training and $2,330$ images for testing. These images cover a diverse range of genders, races, ages, poses, and expressions. The diversity of the Helen dataset is important for well representing the local density modes in the face space, which our method seeks for face beautification. Despite the diverse backgrounds, shapes, poses, and expressions of the input faces, our method manages

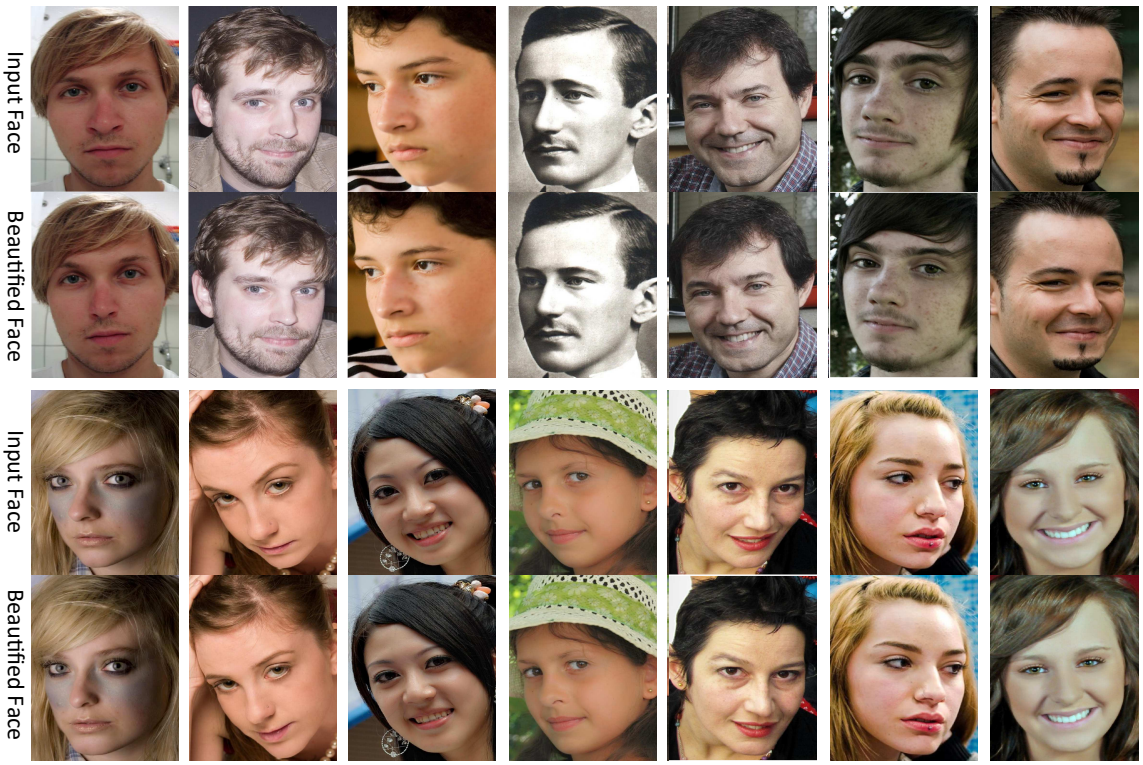


Figure 5.9: **Our Full-face Beautification Results**. These faces are selected to cover a wide range of gender (male and female), pose (frontal and non-frontal), and expression (neutral and non-neutral). Our method is purely unsupervised and does not require any human-annotated facial attractiveness scores for training.

Figure 5.10: **The Comparison of Our Unsupervised Method with the Supervised Method of (Leyvand et al., 2008)**. (*a*): the input facial images. (*b*): the beautified images from (Leyvand et al., 2008). (*c*): the beautified images generated by our method. While the method of (Leyvand et al., 2008) requires human-annotated attractiveness scores for training and only works on frontal portraits, our method does not have these restrictions.

Figure 5.11: **The Comparison of Our Unsupervised Method with the Supervised Method of (Chen et al., 2014)**. *Left*: the input facial images. *Middle*: the beautified images taken from (Chen et al., 2014). *Right*: the beautified images generated by our method. Similar to that of (Leyvand et al., 2008), the method of (Chen et al., 2014) requires human annotations for training and only works on frontal portraits. Our method does not have these restrictions.

to automatically optimise the input facial geometric configurations and produce high-resolution images with improved facial attractiveness. Importantly, the poses and expressions of the input faces are well preserved by our method. Note that neither human-annotated attractiveness scores (Leyvand et al., 2008; Chen et al., 2014; Li et al., 2015) nor 3D face modelling (Liao et al., 2012) are required by our method. Our face beautification results are all generated in a purely unsupervised manner.

Comparisons with Supervised Methods. Here, we compare our method with that of (Leyvand et al., 2008) and (Chen et al., 2014). These two methods only work on frontal portraits and therefore cannot generalise well to non-frontal poses and non-neutral expressions. Besides, they both require human-annotated facial attractiveness scores for training, which unfortunately are difficult to obtain for non-frontal and non-neutral faces that exist in many real-world applications.

Due to the limited number of results provided in (Leyvand et al., 2008) and (Chen et al., 2014), we can only compare with a few facial images taken from the original papers in Fig. 5.10 and 5.11. Both our unsupervised method and the two supervised methods improve the perceived attractiveness of the input faces. While the supervised methods are shown to apply more intense modifications to the eyebrows and eyes, our method preserves more features of these regions. The beautified faces generated by our method appear more similar to the input, which is because our method performs beautification via iterative locality-sensitive averaging. Therefore, it is capable of maintaining some of the input facial features that are normally missing during the supervised optimisation process of (Leyvand et al., 2008) and (Chen et al., 2014).

5.3.2 User Study

To empirically evaluate the effectiveness of our face beautification method, we designed a user study for collecting human preferences towards the original faces or the beautified versions generated by our method. The study consists of two courses, one based on 100 general facial images from the challenging Helen test set (Le et al., 2012) and the other based on 100 more controlled frontal portraits we collect from the Internet. We conducted the study on both sets of images to demonstrate the robustness of our method in handling non-frontal, non-neutral faces in the wild. We generated the beautified results for both sets of images. We recruited 12 male and 12 female human subjects aged 22-35. In each course, we randomly presented pairs of the original and the

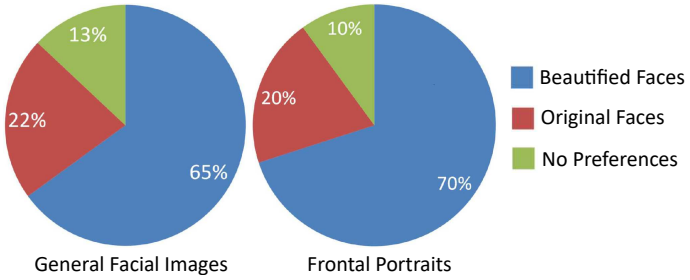


Figure 5.12: **The User Study of Our Method**. *Left*: the percentages of human subjects preferring the original faces, the beautified faces, or without preferences on general facial images. *Right*: the percentages of human preferences on input faces with frontal poses and neutral expressions.

beautified images to each of the subjects, with the order of the images in each pair being randomised. We asked each subject to pick the more attractive face in each pair or to express no preferences. When finished, we obtained $2,347$ and $2,085$ choices for the two courses respectively. To comply with the European General Data Protection Regulation 2016/679 (Voigt and Von dem Bussche, 2017), we asked the subjects for consent to participate in the study and conducted full anonymisation to ensure that no personal identity information can be misused. After forming the overall statistics of preferences as displayed in Fig. 5.12, we also discarded the preferences of the individual subjects to ensure that no personal preferences towards facial attractiveness evaluation can be inferred.

As shown in Fig. 5.12, the subjects in both courses prefer the full-face beautified images over the originals. Remarkably, the percentages of subjects preferring our results on the much more challenging general facial image course, which includes non-frontal and non-neutral faces, are comparable to that on the frontal portraits (70% versus 65%). This demonstrates the robustness of our method for enhancing non-frontal and non-neutral faces, which are very common in many real-world applications. To our knowledge, this is the first time in the field that a method has been shown to be capable of beautifying such challenging faces.

5.3.3 User-controlled Beautification Results

Beyond full-face beautification without user intervention, our method allows users to prescribe a beautification weight in $[0, 1]$ for each individual facial part. The smaller the weight, the better preserved the shape of the corresponding part. Fig. 5.13 shows some user-customised face beautification results, for which we allow users to preserve one or more facial parts while fully

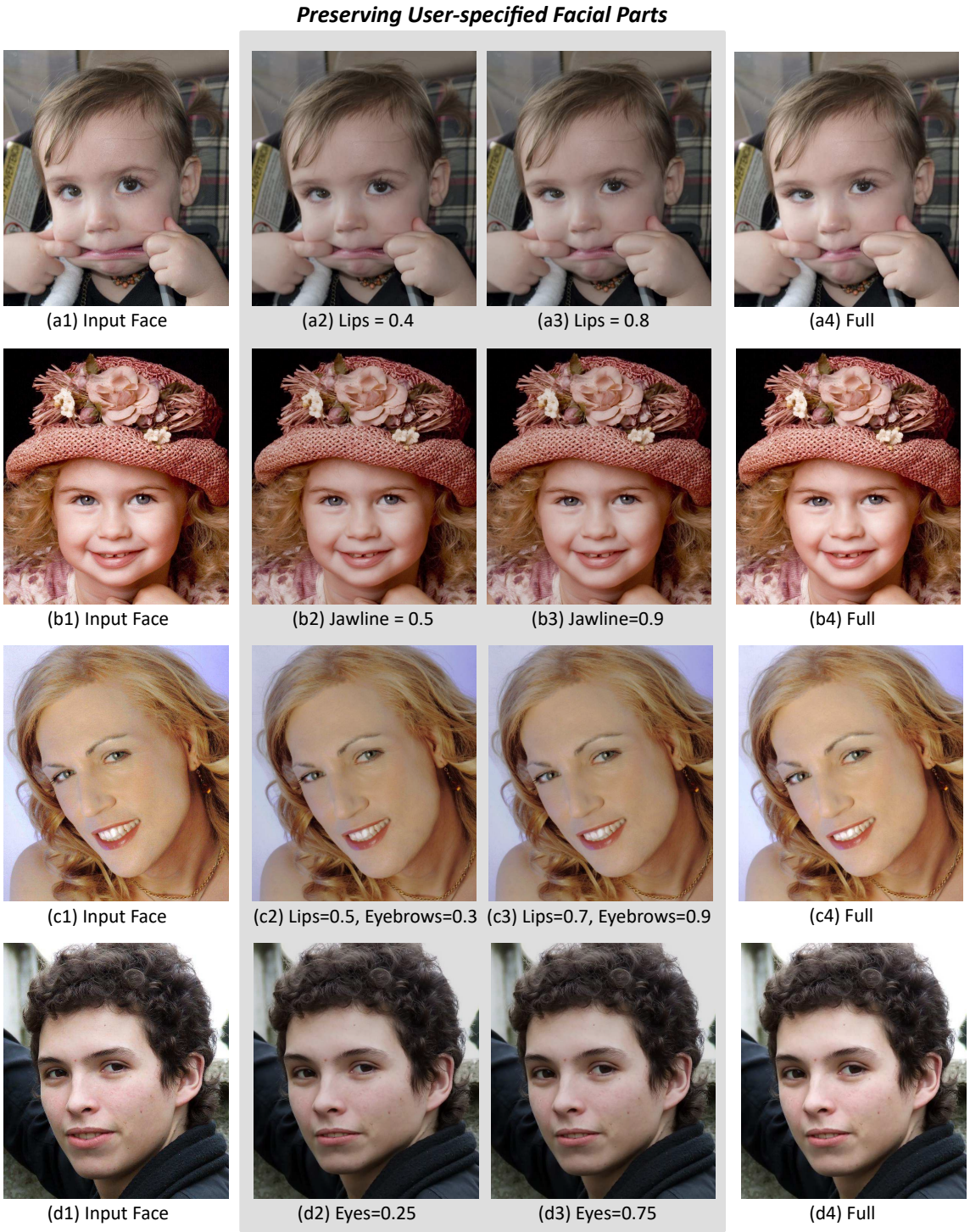


Figure 5.13: **Our User-controlled Face Beautification Results**. Here, we allow users to preserve one or more facial parts while fully beautifying the remaining. On each row, we show the input facial image on the first column and the full-face beautified images on the fourth column. We show the intermediate beautified images on the second and third columns, with the user-customised beautification weights of the facial parts being preserved. The beautification weights of the parts not being preserved are set to the largest value of 1.0.

Table 5.1: **The Run Time of Our Method for Beautifying a Typical Input Facial Image of Resolution** 1280×960. The steps (b)-(d) are our contributions in this work.

	Operation	Run time
Step (a)	Facial Landmarks Detection	0.451723s
Step (b)	**Facial Metric Computation**	**0.000117s**
Step (c)	**Facial Metric Optimisation**	**0.190622s**
Step (d)	**Facial Landmarks Reshaping**	**0.002041s**
Step (e)	Facial Image Warping	0.323160s
Total	The Whole Process	0.967839s

beautifying the remaining. As shown on the first row, our method generates natural and smooth transitions between the shape of the input mouth and that of the fully beautified version, when users change the beautification weights of the outer and inner lips from 0.0 (i.e. no beautification) to 0.4, 0.8, and 1.0 (full beautification). Our method works equally well for allowing users to control the beautification level of the jawline (on the second row) and eyes (on the fourth row) of an input face. More challenging is the case when users want to preserve more than one facial part, as shown in the third row of Fig. 5.13. Still, our method generates intermediate beautified images that naturally interpolate the original and the fully enhanced eyebrows and mouth.

5.3.4 Computational Cost

We list the run time of each computation step of our face beautification method in Table 5.1. The time is measured for processing a typical facial image of resolution 1280×960 on a laptop with an Intel(R) Core i7-6500U 2.5GHZ CPU and an 8GB RAM. Overall, the whole beautification process takes less than 1 second to finish, which allows for continuous user interactions with low latency. The steps of facial metric computation (b), facial metric beautification (c), and facial landmarks reshaping (d) are extremely efficient because (b) mainly involves the inverse of a 3×3 matrix, (c) is accelerated using a KD-tree structure, and (e) is simply a linear projection. While the step of facial image warping (e) is the most time-consuming, it is still reasonably fast given the high resolution of the input image. Note that the steps of facial landmarks detection (a) and facial image warping (e) are not the focus of this work, and they could be replaced by other faster methods in the future.

5.4 Summary

In this chapter, we have proposed a metric representation for purely unsupervised facial shape beautification, without requiring any human-annotated facial attractiveness scores for training. We have proposed to formulate face beautification as the process of iteratively pulling the metric representation of an input face towards a local nearby density mode in the metric space of faces. Benefiting from the orthogonal projection nature of our metric representation, it frees the beautification process from the nuisance affine transformations of facial landmarks and naturally groups facial landmarks into individual facial parts for user control. It also significantly simplifies the reconstruction of beautified facial landmarks as linear projection, which is very efficient and guaranteed to be globally optimal. Due to the locality of our iterative face averaging method, the beautified version of an input face faithfully preserves the original pose and expression, which have been confirmed by many examples shown in the work. Our method has been shown to be capable of beautifying both frontal portraits and general facial images that contain a wide range of non-frontal poses and non-neutral expressions, which is beyond the ability of the current supervised face beautification methods. On top of this, our method enables users to flexibly prescribe beautification weights so that certain facial parts can be preserved more while others can be enhanced more. As the speed of our method is interactive, it allows users to continuously customise the beautification weights until satisfactory results are obtained.

Chapter 6

Human Shape and Pose Modelling

In Chapter 5, we have studied the problem of facial shape synthesis using our proposed metric representation framework. In this chapter, we plan to generalise the singular task of shape synthesis to the modelling of shapes and poses for humans. The motivation is that virtual human synthesis has recently become a very active research topic in computer graphics, which has broad applications in computer games and virtual reality. As an extension work of this thesis, our key idea is learning the low-dimensional probability manifolds of human shapes and poses in the auto-encoding rather than the original geometry space with GANs. Because the underlying dimensions of the manifolds are independent of the ambient embedding space, the learning task can be made considerately easier for synthesising higher-quality human samples.

6.1 Introduction

3D human body shapes and poses are ubiquitous in computer games and animations. Modelling their distributions is a fundamental building block for automatically synthesising realistic-looking human characters and animations.

The drawback of linear subspace methods such as the SMPL model (Loper et al., 2015) is that random sampling far from the centre (i.e. the average human shape and pose) of the resulting Gaussian distributions would produce non-realistic humans (Kanazawa et al., 2018). This is because the distributions of real human shapes and poses are more likely to be locally supported on non-linear manifolds, instead of on the full subspaces where the Gaussian distributions are

globally supported on. Also, the SMPL model is mainly used for reconstructing human shapes and poses from images or videos (Kanazawa et al., 2018; Bogo et al., 2016; Lassner et al., 2017), rather than for synthesising virtual humans without any guidance. In contrast, we focus on generating humans via random sampling in this study and pursue the task of guided reconstruction in the future work.

The more recent work on generative adversarial networks (GANs) for non-linear distribution modelling (Goodfellow et al., 2014) still cannot produce satisfactory results, because they embed the shape and pose manifolds directly in the high-dimensional geometry spaces (Kanazawa et al., 2018). The training can be difficult because measuring the overlap of two manifolds (i.e. the real and the generated) in high-dimensional spaces is difficult.

In this work, we propose to learn the distributions of real human shapes and poses using two separate GANs, not in the original high-dimensional geometry spaces but in the more representative low-dimensional latent spaces discovered by a shape and a pose auto-encoder network respectively. The motivation is that the dimensions of the real shape and pose manifolds should be independent of the ambient spaces. As a result, measuring their overlaps with the generated shape and pose manifolds can be made easier in a much lower-dimensional ambient space. Therefore, we train a shape encoder (similarly a pose encoder) to embed real human shapes into a low-dimensional hidden space, while training a shape decoder (similarly a pose decoder) to reconstruct the input. We train a shape generator (similarly a pose generator) to transform a standard Gaussian distribution into this space, in which a shape discriminator (similarly a pose discriminator) is also trained to approximate the distribution distance for the generator to minimise.

We propose the two following contributions in this work:

- We propose to learn the non-linear manifolds of real human shapes and poses separately in two respective low-dimensional auto-encoding spaces, rather than in the original high-dimensional geometry spaces. We note that there is no sharing of parameters between shape and pose modelling in this study.
- We demonstrate the capacity of our learned priors by generating high-quality human shapes and poses via random sampling. We release our source code* to facilitate the synthesis of

*https://drive.google.com/open?id=1y-aPe8FGzetxnY3FpSESci3U59KgQSUJ

realistic humans in real-time (around 5ms using a GTX 1080 graphics card).

6.2 DSPP: Deep Shape and Pose Priors of Humans

We focus on modelling the probability distributions of real human body shapes $x \sim \boldsymbol{p}(x)$ and real human body poses $y \sim \boldsymbol{p}(y)$. In computer gaming and animation, it is typical to represent a human shape $x \in \mathbb{R}^{N\times 3}$ as a 3D point cloud with a given mesh topology, and a human pose $y \in \mathbb{R}^{M\times 3}$ as an array of 3D joint Euler angles. Combining the two using skinning techniques can produce a deformed human shape in the given pose (Wang et al., 2015).

The challenge of modelling $\boldsymbol{p}(x)$ and $\boldsymbol{p}(y)$ is that they are high-dimensional distributions but can only be specified as the empirical distributions of scanned human shapes and motion-captured human poses in practice. That is, sampling from them is equivalent to sampling from a shape dataset and a pose dataset respectively. As a result, our task is approximating them using continuous distributions $\hat{\boldsymbol{p}}(x)$ and $\hat{\boldsymbol{p}}(y)$ so that the drawn samples are visually similar to that from the given datasets.

The overview of our method is illustrated in Fig. 6.1. The key is that we assume $\hat{\boldsymbol{p}}(x)$ and $\hat{\boldsymbol{p}}(y)$ to be locally supported on two low-dimensional manifolds respectively, whose dimensions are independent of the ambient spaces the manifolds are embedded in. Our method departs from the previous work on linear subspaces that assume both distributions to be Gaussian with a global support (Loper et al., 2015). It is also in contrast with the recent work on generative distribution modelling that embeds the manifolds in the original high-dimensional geometry space. Particularly, our method learns some low-dimensional ambient spaces and the manifolds embedded within, which is shown to produce higher-quality samples.

6.2.1 Auto-encoding Ambient Spaces

To find the low-dimensional ambient spaces for generative distribution modelling, we learn two pairs of deep encoders $\{f_{\text{shape}}, f_{\text{pose}}\}$ and decoders $\{h_{\text{shape}}, h_{\text{pose}}\}$, by minimising the two follow-

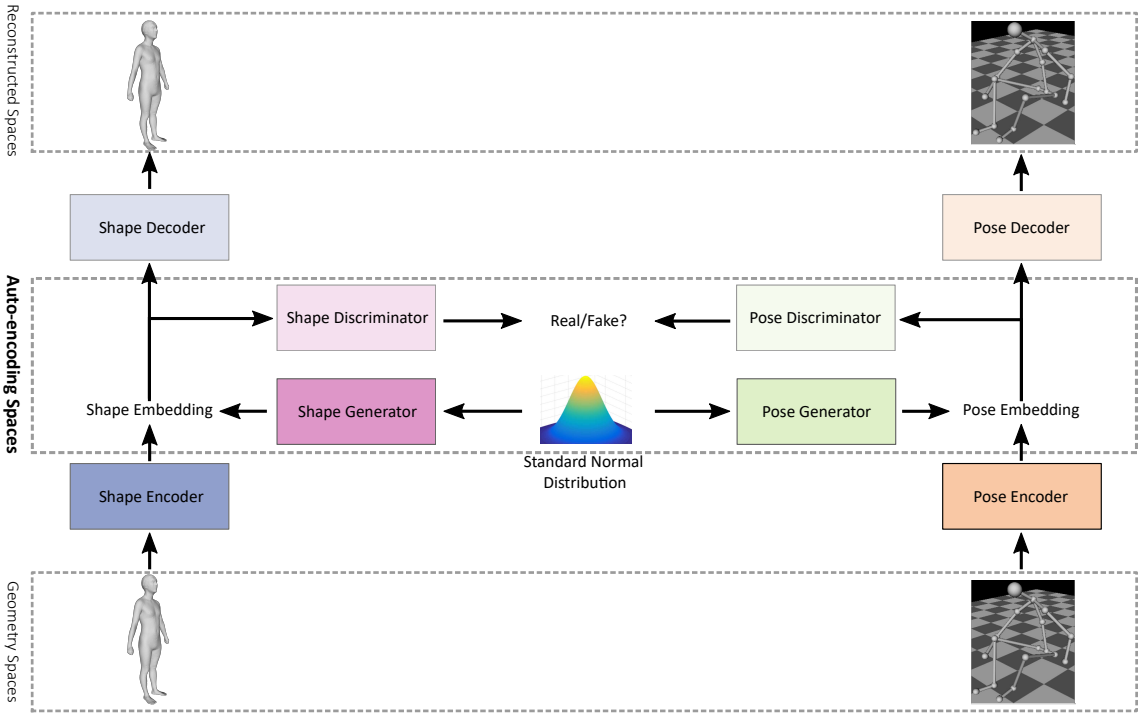


Figure 6.1: **The Overview of Our Human Modelling System**. We train a pair of shape encoder/decoder and a pair of pose encoder/decoder so that we can learn the non-linear manifolds of human shape and pose in the low-dimensional auto-encoding ambient spaces. Synthesizing realistic human shapes and poses amounts to sampling from a standard normal distribution, then applying the corresponding generators, and finally applying the corresponding decoders. We note that the shape and pose modelling branches are separately trained with no sharing of parameters between them.

ing reconstruction losses respectively:

$$l(x, f_{\text{shape}}, h_{\text{shape}}) = \boldsymbol{E}_{x\sim\boldsymbol{p}(x)}\|x - h_{\text{shape}}(f_{\text{shape}}(x))\|_F^2 \tag{6.1}$$

$$l(y, f_{\text{pose}}, h_{\text{pose}}) = \boldsymbol{E}_{y\sim\boldsymbol{p}(y)}\|y - h_{\text{pose}}(f_{\text{pose}}(y))\|_F^2 \tag{6.2}$$

where the expectations of the squared Frobenius norms[†] are taken with respect to the empirical shape and pose distributions, $\boldsymbol{p}(x)$ and $\boldsymbol{p}(y)$, respectively.

We denote $z_x = f_{\text{shape}}(x)$ and $z_y = f_{\text{pose}}(y)$ as the learned hidden representations of real human shapes and poses respectively. This allows us to obtain $z_x \sim \boldsymbol{p}(z_x)$ and $z_y \sim \boldsymbol{p}(z_y)$ as the low-dimensional empirical distributions of real human shape and pose representations. Because we can use a much smaller dimension for z_x and z_y respectively, our task of approximating the original high-dimensional empirical distributions $\boldsymbol{p}(x)$ and $\boldsymbol{p}(y)$ can be greatly simplified to model $\boldsymbol{p}(z_x)$ and $\boldsymbol{p}(z_y)$ instead.

6.2.2 Generative Modelling in the Auto-encoding Spaces

Now, we seek to find a pair of continuous distributions $\hat{\boldsymbol{p}}(z_x)$ and $\hat{\boldsymbol{p}}(z_y)$ so that they can be made close to their empirical distribution counterparts, by minimising some distribution distance between them. Assuming the continuous distributions to be Gaussian and taking the distance as the Kullback−Leibler divergence, as done in the previous work on linear subspace methods, recover the globally supported Gaussian distributions that produce non-realistic samples far from the centres.

Instead, we assume $\hat{\boldsymbol{p}}(z_x)$ and $\hat{\boldsymbol{p}}(z_y)$ to be only locally supported on the low-dimensional human shape and pose manifolds, which are embedded in the auto-encoding shape and pose ambient spaces respectively. Therefore, we leverage the state-of-the-art GANs for discovering such non-linear manifolds (Goodfellow et al., 2014). We train a shape generator $\varphi_{\text{shape}} : z \to z_x$ and a pose generator $\varphi_{\text{pose}} : z \to z_y$ that embed a random sample $z \sim \mathcal{N}(0, \boldsymbol{I})$ from a low-dimensional standard Gaussian distribution into the shape and pose ambient spaces respectively. This way, synthesising human shapes and poses from the embedded manifolds is equivalent to sampling z and then applying the two generator functions respectively.

[†] $\|X\|_F^2 = \text{trace}(X^T X)$

To fit the embedded shape and pose manifolds to the empirical distributions $\boldsymbol{p}(z_x)$ and $\boldsymbol{p}(z_y)$, we take the popular discrimination loss:

$$l(\hat{\boldsymbol{p}}(z_x), \boldsymbol{p}(z_x)) = -\boldsymbol{E}_{z\sim\mathcal{N}(0,\boldsymbol{I})}[\log\psi_{\text{shape}}(\varphi_{\text{shape}}(z))] \tag{6.3}$$

$$l(\hat{\boldsymbol{p}}(z_y), \boldsymbol{p}(z_y)) = -\boldsymbol{E}_{z\sim\mathcal{N}(0,\boldsymbol{I})}[\log\psi_{\text{pose}}(\varphi_{\text{pose}}(z))] \tag{6.4}$$

where $\psi_{\text{shape}} : z_x \rightarrow (0, 1)$ and $\psi_{\text{pose}} : z_y \rightarrow (0, 1)$ are the separately trained discriminator functions that tell whether a given shape and a pose are real samples or not: near-1 probabilities mean real and near-0 probabilities mean synthetic. Minimising (6.3) and (6.4) amounts to finding the shape and pose generators that produce synthetic samples indistinguishable from the real ones according to the corresponding discriminators (Goodfellow et al., 2014).

6.2.3 Datasets, Architectures, and Training

To validate the effectiveness of our method, we train our human shape and pose distribution modelling system on the MPII Human Shape dataset (Pishchulin et al., 2017) and the SFU Motion Capture dataset (*SFU Motion Capture dataset*, n.d.). The former contains 4,308 3D human body shapes registered to a common mesh topology, with each shape consisting of 6,449 surface points. The latter provides 3D human motion capture clips covering various activities such as walking, running, dancing, and interactions. We extract over 100,000 poses from these clips using a regular sampling rate of 4 frames/second. We represent each pose using the 3D Euler angles of 20 body joints defined in (Pishchulin et al., 2017), excluding that of the Hips root joint as it describes global rotations.

We illustrate the neural network architectures of our shape and pose subsystems in Fig. 6.2 and 6.3 respectively. We also show the baseline shape and pose discriminator architectures that work in the original geometry spaces in Fig. 6.4, where we use the max-pooling operator to aggregate point features for global context modelling (Qi, Su, Mo and Guibas, 2017*b*). We use the Parametric Rectified Linear Unit (PReLU) activation function for the encoders, decoders, and generators (He et al., 2015), while using the Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) for the discriminators. Importantly, we follow the method of (Miyato et al., 2018) to normalise the spectral norm (SN) of each linear transformation in the discriminators to be 1, which effectively stabilises the training of GANs. In each of 100,000 training cycles, we first randomly sample a pair of real

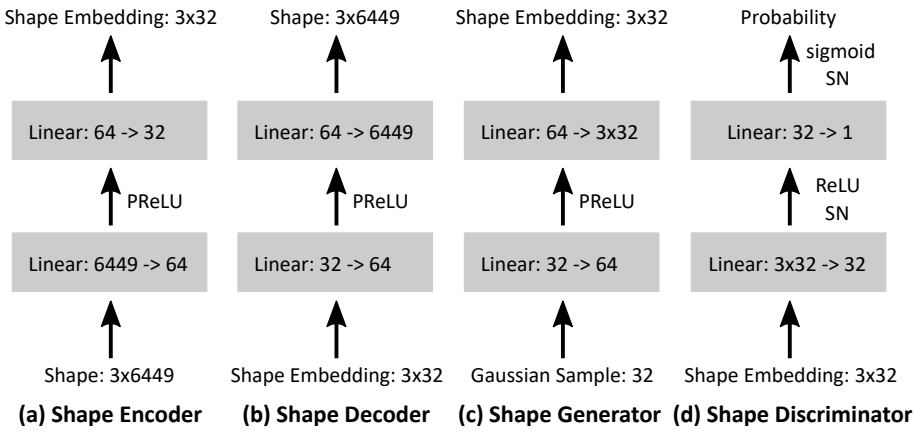


Figure 6.2: **The neural network architectures of our proposed shape encoder, shape decoder, shape generator, and shape discriminator**.

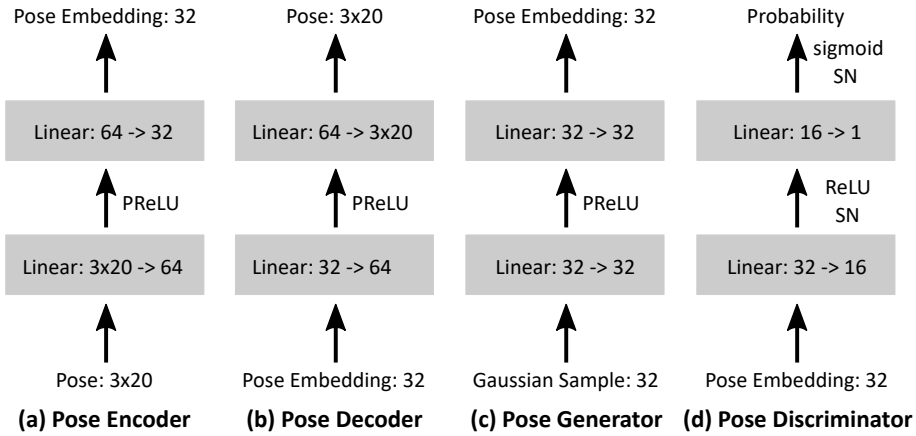


Figure 6.3: **The neural network architectures of our proposed pose encoder, pose decoder, pose generator, and pose discriminator**.

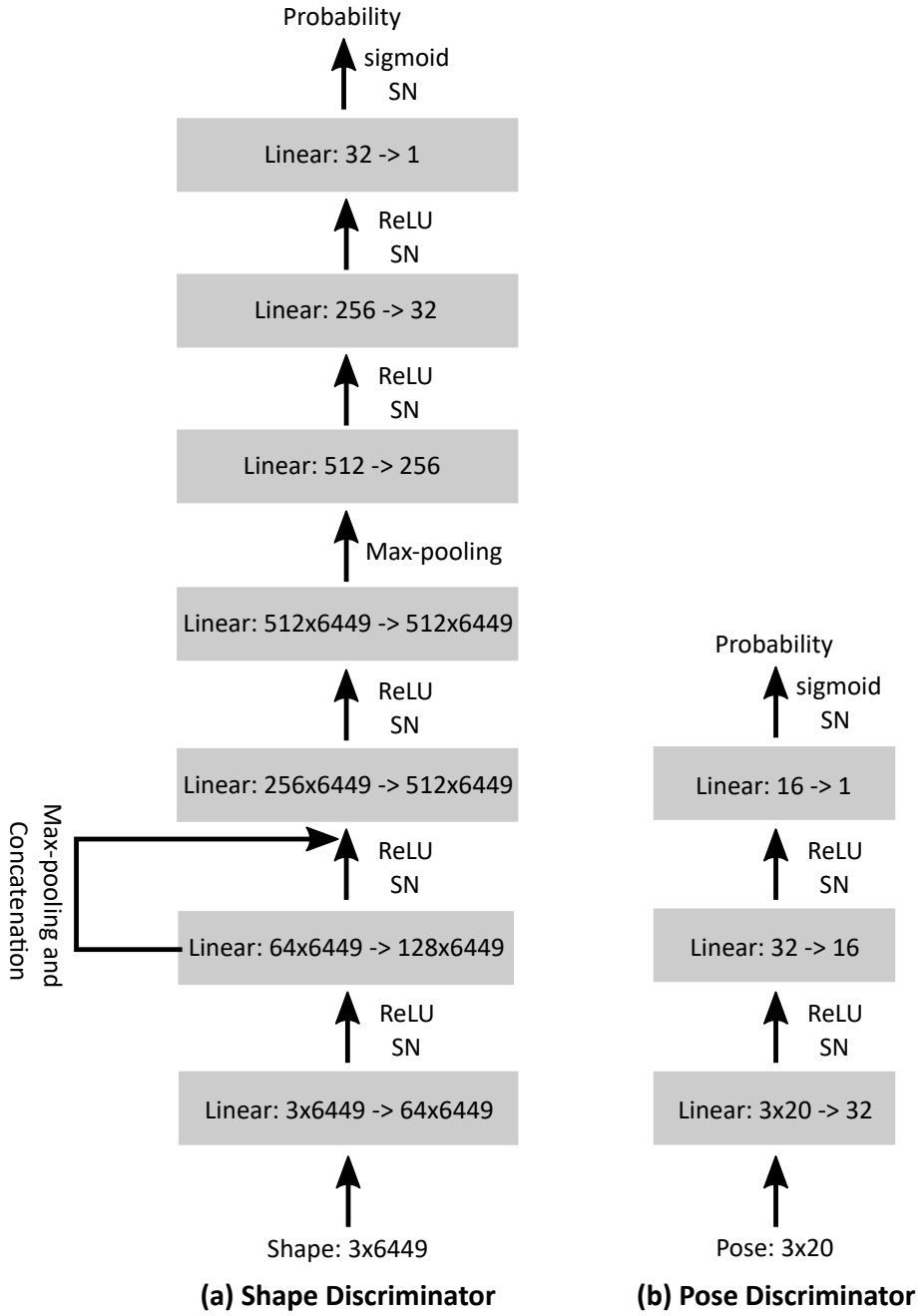


Figure 6.4: **The architectures of the baseline shape and pose discriminators that work in the original geometry ambient spaces**.

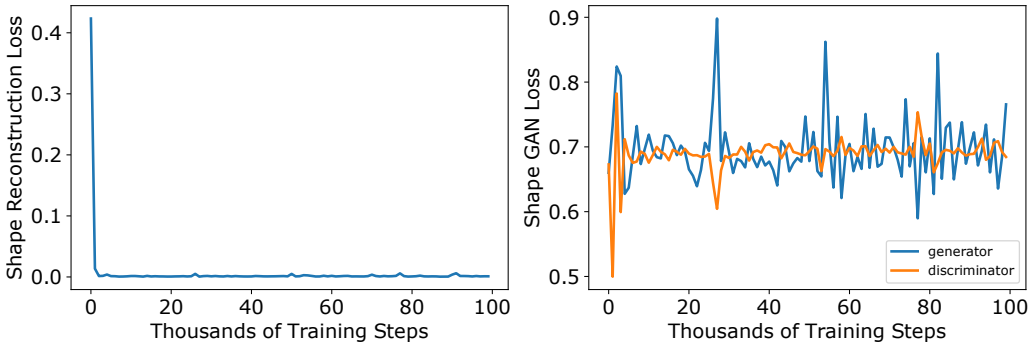


Figure 6.5: **The loss of our shape encoder, shape decoder, shape generator, and shape discriminator**.

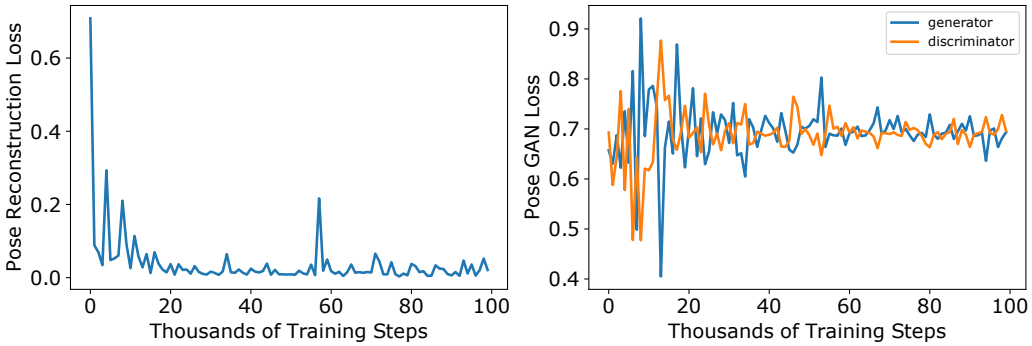


Figure 6.6: **The loss of our pose encoder, pose decoder, pose generator, and pose discriminator**.

shape and pose to train the corresponding encoders and decoders. We then train the discriminators using the updated embeddings and a pair of synthetic samples computed from the respective generators. Finally, we train the two generators based on the corresponding updated discriminators. We use the Adam optimiser with the default settings for training (Kingma and Ba, 2014).

We implement our system in PyTorch (V1.1) on a PC with a GTX 1080 graphics card with 8GB graphics memory. Each training step takes around 70ms, and generating a pair of shape and pose takes around 5ms. Our code is publicly available from this link: https://drive.google.com/open?id=1y-aPe8FGzetxnY3FpSESci3U59KgQSUJ

6.3 Results

We show the training losses of our proposed human shape and pose modelling subsystems in Fig. 6.5 and 6.6 respectively. As the training progresses, both the shape and pose reconstruction losses decrease rapidly. This shows that the shape and pose encoders effectively compress the original

Figure 6.7: **Human Shape Comparison**. Comparison of our randomly sampled human shapes with that sampled from the baseline method of learning discrimination in the original shape space. We use red rectangles to indicate samples that do not look realistic.

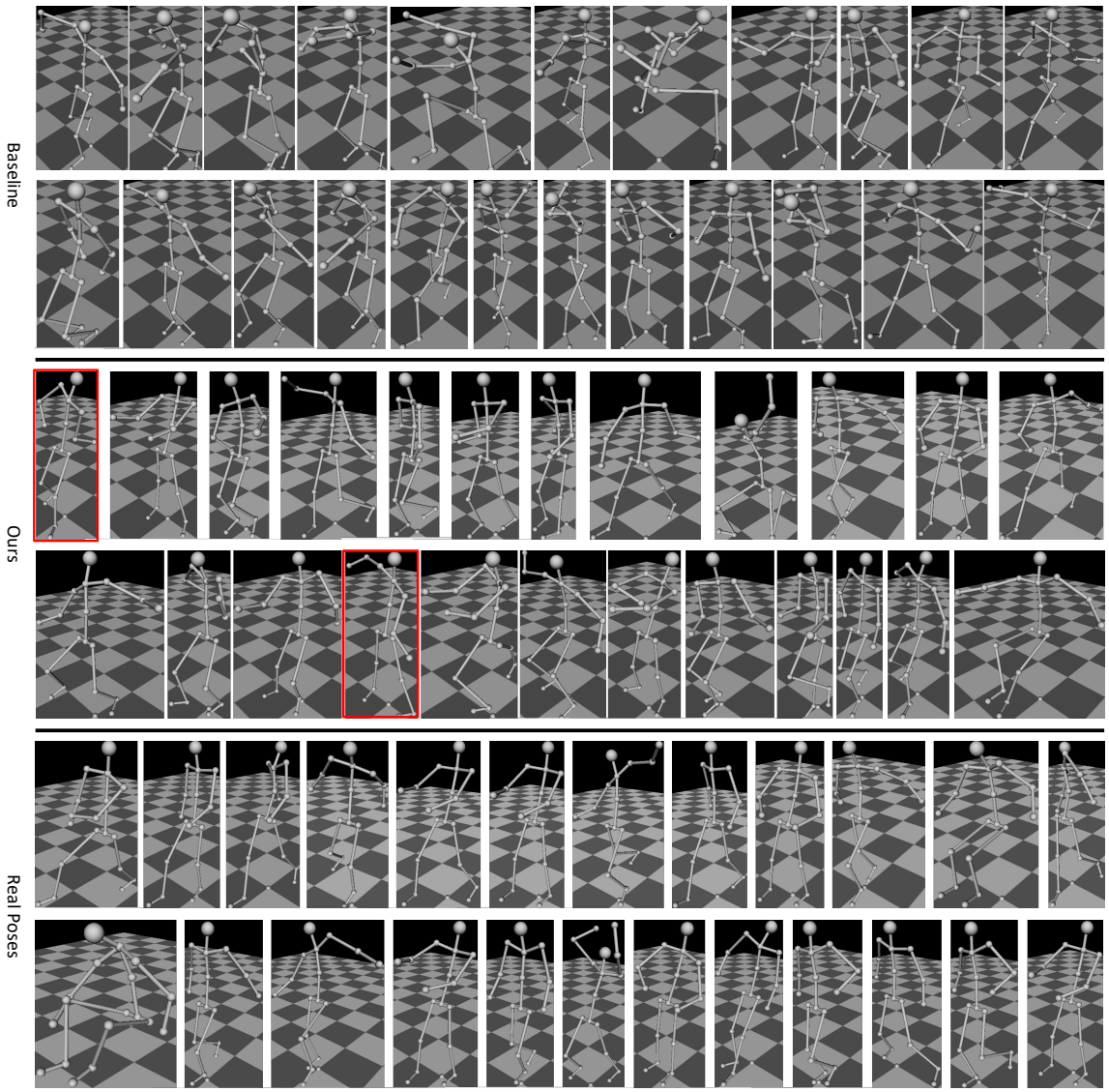


Figure 6.8: **Human Pose Comparison**. Comparison of our randomly sampled human poses with that sampled from the baseline method of learning discrimination in the original pose space. We use red rectangles to indicate our samples that do not look realistic.

Table 6.1: **The statistics (mean and standard deviation) of the distances from the generated samples to the nearest true samples.**

	Human Shape	Human Pose
Baseline	3.42 ± 1.12	2.79 ± 1.56
Ours	$\mathbf{2.68 \pm 1.32}$	$\mathbf{2.01 \pm 1.54}$

geometry input into the more representative spaces, from which the corresponding decoders successfully reconstruct the input. The fluctuations of the shape and pose GAN losses are desirable, which indicate that the generators are continuously learning to produce samples that cannot easily be distinguished by the corresponding discriminators.

We compare our randomly sampled human shapes with that sampled from the method of using the baseline shape discriminator (Fig. 6.4, left) in Fig. 6.7. Our synthesised shapes look both realistic and diverse, resembling closely with the samples from the ground-truth dataset (Pishchulin et al., 2017). In comparison, the baseline method produces noticeable noises around the head, chest, and feet regions as indicated by the red rectangles, which are caused by the insufficient capacity of the baseline discriminator that fails to capture fine-scale surface details. We also compare our randomly sampled human poses with that sampled from the method of using the baseline pose discriminator (Fig. 6.4, right) in Fig. 6.8. Similarly, our generated poses look considerably more realistic than the baseline results. Together, these results validate the effectiveness of modelling shape and pose manifolds in the auto-encoding spaces.

To quantitatively compare the realism of our generated samples with the baseline samples, we compute the distances from them to their nearest true samples in the shape (Pishchulin et al., 2017) and pose (*SFU Motion Capture dataset*, n.d.) datasets respectively, as summarised in Table 6.1. The distance between two shapes is computed as the Frobenius norm of the difference of their coordinates, while the distance between two poses is computed as the Frobenius norm of the difference of their Euler angles. We use 100,000 randomly sampled shapes and poses for comparison. It can be seen that the average distances from our generated shapes and poses to the true samples are both smaller than from the baseline samples to the true samples. This shows that our samples are geometrically closer to the real human scans and captures. We note that the nearest sample distance is an approximate way to measure the visual realism of automatically generated samples, and it may be enhanced by using controlled user studies in the future work.

6.4 Summary

In this chapter, we introduced the idea of modelling the non-linear manifolds of human shapes and poses in the auto-encoding ambient spaces. We discovered the spaces by learning a pair of encoder and decoder for human shapes and poses respectively. The low-dimension of such spaces allow us to more effectively learn human shape and pose manifolds, using the powerful non-linear distribution modelling GAN. The learned manifolds, as embedded in the auto-encoding ambient spaces, allow for the synthesis of realistic human shapes and poses. Our results showed that our method produces higher-quality samples, comparing with the method of distribution modelling in the original geometry spaces.

Chapter 7

Conclusion and Future Work

In this thesis, we have proposed a unified metric framework for shape analysis and synthesis, which targets at three important yet diverse applications in computer graphics: facial shape beautification, mesh saliency detection, and non-rigid shape matching. We have framed the central theme of our framework as representing 2D/3D shapes as semantically informative metrics that capture the global geometric features of shapes for analysis and synthesis. The global nature of our metric representations has successfully allowed us to formulate the three problems in a coherent global optimisation setup, where the mathematical goal is to leverage the eigenvectors of the metrics for analysis and synthesis. Concretely, we have formulated facial shape synthesis as linear projection onto the eigenvectors of a facial orthogonal projection metric. We have also formulated saliency detection as computing the sparse principal eigenvector of the feature distance metric of a mesh and non-rigid shape matching as computing the Laplacian eigenvectors of the feature distance metric that is learned by deep learning. The successes achieved for all three applications demonstrate the usefulness of our metric representations for shape analysis and synthesis. As an extension work, we have also generalised shape synthesis to the joint modelling of human shapes and poses for virtual human synthesis. By learning the manifolds of shapes and poses in a low-dimensional auto-encoding space, the generated samples are higher-quality than that learned in the original geometry space. In the following, we discuss the future directions for each of these work.

7.1 Mesh Saliency Detection

We have selected 50 commonly used graphics meshes for our 3D eye fixation dataset construction. In the future, we could scale the construction to the SHREC2007 dataset (*SHREC'2007 watertight mesh database*, n.d.) so that we can have each of the 400 meshes annotated with 3D eye fixations and Schelling saliency values (Chen et al., 2012). This would greatly facilitate the large-scale benchmarking of saliency detection methods for further progress. Implementing more methods (e.g. of (Feixas et al., 2009; Zhao et al., 2016; Jeong and Sim, 2017)) will also help this.

Our assumption of saliency is a bottom-up computational approximation to the pre-attentive mechanism of the human visual system. As a result, it may fail to capture a few visually salient but not necessarily rare regions of a mesh, such as the chest of the Armadillo and the face of the Teddy, as shown in Figure 3.5. To capture these challenging regions, more high-level cues like symmetries, segmentation, and semantic annotations are expected to be helpful.

7.2 Unification of Saliency Detection and Shape Matching

Currently, our system requires dense point-to-point correspondences to enforce the intra-category consistency property, which have very limited availability and are difficult to label (Anguelov et al., 2005; Vlasic et al., 2008; Bogo et al., 2014). This may be partly addressed with sparse segment correspondences (Ovsjanikov et al., 2012), but a more favourable bootstrap solution would be to compute less accurate matchings for improvement with target tasks jointly and iteratively.

The Laplacian spectral embeddings (Rustamov, 2007) and our saliency-induced ones represent the two extreme ends of discrimination-invariance trade-off, with the former proven to be the smoothest and the latter proven to be the most localised. Therefore, our embeddings lack fine-grained discrimination for non-salient points. This is why we incorporate our embeddings into the model-based SDP and the learning-based MLP methods for shape matching. In between the Laplacian spectral embeddings and ours, there would be an optimal discrimination-invariance trade-off that takes both salient and non-salient points into consideration. Finding the optimal solution depends on the applications and is a future direction.

7.3 Facial Shape Beautification

The computational bottlenecks of our face beautification method are the steps of facial landmarks detection and facial image warping. These are not our focus in this work and can be significantly accelerated by using other faster methods, such as the one of (Zhang et al., 2014) for landmarks detection and that of (Liao et al., 2014) for image warping.

Currently, to ease user control in face beautification, we have allowed users to customise beautification weights on the level of facial parts (e.g. the jawline and the eyes). An interesting direction of research is automatically learning these weights from high-level personal traits such as gender, age, and race.

We have formulated facial shape beautification as an iterative locality-sensitive averaging process in the face space. We have specified the locality of face averaging using a scale parameter γ, which has the advantage of producing minimum changes to input faces so that major facial features are well preserved. In the future, the choice of γ could be optimised based on user preferences towards facial attractiveness. Non-local faces could also be incorporated in the averaging process to suggest more diverse, user-satisfied face changes for beautification.

We have focused on the beautification of 2D facial shapes in this work. Still, our facial reshaping metric and locality-sensitive averaging method are independent of the specific coordinate dimension, which can be seamlessly adapted for the beautification of 3D facial meshes (Liao et al., 2012).

7.4 Human Shape and Pose Synthesis

As a future work, we are interested in learning to realistically deform the synthesised shapes using the synthesised poses. This will open the door of automatic human character synthesis in applications such as animation and gaming. Although the method of Loper et al. (2015) permits such applications, it relies on the traditional skinning technique and could result in non-realistic deformations. The work of Bailey et al. (2018) that automatically learns deformations is worthy exploration.

References

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J. and Davis, J. (2005), 'Scape: shape completion and animation of people', *ACM Transactions on Graphics* **24**(3), 408–416.

Argyriou, A., Evgeniou, T. and Pontil, M. (2007), Multi-task feature learning, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 41–48.

Bailey, S. W., Otte, D., Dilorenzo, P. and O'Brien, J. F. (2018), 'Fast and deep deformation approximations', *ACM Transactions on Graphics* **37**(4), 119.

Belkin, M. and Niyogi, P. (2002), Laplacian eigenmaps and spectral techniques for embedding and clustering, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 585–591.

Belkin, M., Sun, J. and Wang, Y. (2008), Discrete laplace operator on meshed surfaces, *in* 'Proceedings of the twenty-fourth annual symposium on Computational geometry', pp. 278–287.

Belongie, S., Malik, J. and Puzicha, J. (2001), Shape context: A new descriptor for shape matching and object recognition, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 831–837.

Berman, A. and Plemmons, R. J. (1994), *Nonnegative matrices in the mathematical sciences*, Society for Industrial and Applied Mathematics.

Billinghurst, M., Clark, A., Lee, G. et al. (2015), 'A survey of augmented reality', *Foundations and Trends® in Human–Computer Interaction* **8**(2-3), 73–272.

Blanz, V. and Vetter, T. (1999), A morphable model for the synthesis of 3d faces, *in* 'Proceedings of the ACM SIGGRAPH', pp. 187–194.

Blanz, V., Vetter, T. et al. (1999), A morphable model for the synthesis of 3d faces., *in* 'Proceedings of ACM SIGGRAPH', pp. 187–194.

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J. and Black, M. J. (2016), Keep it smpl: Automatic estimation of 3d human pose and shape from a single image, *in* 'Proceedings of European Conference on Computer Vision', Springer, pp. 561–578.

Bogo, F., Romero, J., Loper, M. and Black, M. J. (2014), Faust: Dataset and evaluation for 3d mesh registration, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 3794–3801.

Borji, A., Cheng, M.-M., Jiang, H. and Li, J. (2015), 'Salient object detection: A benchmark', *IEEE Transactions on Image Processing* **24**(12), 5706–5722.

Borji, A. and Itti, L. (2013), 'State-of-the-art in visual attention modeling', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(1), 185–207.

Boscaini, D., Eynard, D., Kourounis, D. and Bronstein, M. M. (2015), 'Shape-from-operator: Recovering shapes from intrinsic operators', *Computer Graphics Forum* **34**(2), 265–274.

Boscaini, D., Masci, J., Melzi, S., Bronstein, M. M., Castellani, U. and Vandergheynst, P. (2015), 'Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks', *Computer Graphics Forum* **34**(5), 13–23.

Boscaini, D., Masci, J., Rodolà, E. and Bronstein, M. (2016), Learning shape correspondence with anisotropic convolutional neural networks, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 3189–3197.

Boscaini, D., Masci, J., Rodolà, E., Bronstein, M. M. and Cremers, D. (2016), 'Anisotropic diffusion descriptors', *Computer Graphics Forum* **35**(2), 431–441.

Botsch, M., Kobbelt, L., Pauly, M., Alliez, P. and Lévy, B. (2010), *Polygon mesh processing*, AK Peters/CRC Press.

Bottou, L. (2010), Large-scale machine learning with stochastic gradient descent, *in* 'COMPSTAT'2010', pp. 177–186.

Boyer, E., Bronstein, A. M., Bronstein, M. M., Bustos, B., Darom, T., Horaud, R., Hotz, I., Keller, Y., Keustermans, J., Kovnatsky, A., Litman, R., Reininghaus, J., Sipiran, I., Smeets, D., Suetens, P., Vandermeulen, D., Zaharescu, A. and Zobel, V. (2011), Shrec 2011: Robust feature detection and description benchmark, *in* 'Proceedings of the 4th Eurographics Conference on 3D Object Retrieval', Eurographics Association, pp. 71–78.

Brand, M. and Pletscher, P. (2008), A conditional random field for automatic photo editing, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 1–7.

Bronstein, A. M., Bronstein, M. M., Guibas, L. J. and Ovsjanikov, M. (2011), 'Shape google: Geometric words and expressions for invariant shape retrieval', *ACM Transactions on Graphics* **30**(1), 1.

Bronstein, A. M., Bronstein, M. M. and Kimmel, R. (2008), *Numerical geometry of non-rigid shapes*, Springer Science & Business Media.

Bruce, N. and Tsotsos, J. (2005), Saliency based on information maximization, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 155–162.

Cao, C., Bradley, D., Zhou, K. and Beeler, T. (2015), 'Real-time high-fidelity facial performance capture', *ACM Transactions on Graphics* **34**(4), 46.

Castellani, U., Cristani, M., Fantoni, S. and Murino, V. (2008), 'Sparse points matching by combining 3d mesh saliency with statistical descriptors', *Computer Graphics Forum* **27**(2), 643–652.

Chen, F., Xu, Y. and Zhang, D. (2014), 'A new hypothesis on facial beauty perception', *ACM Transactions on Applied Perception* **11**(2), 8:1–8:20.

Chen, X., Golovinskiy, A. and Funkhouser, T. (2009), 'A benchmark for 3d mesh segmentation', *ACM Transactions on Graphics* **28**(3), 1–12.

Chen, X., Saparov, A., Pang, B. and Funkhouser, T. (2012), 'Schelling points on 3d surface meshes', *ACM Transactions on Graphics* **31**(4), 1–12.

Chen, Y., Shen, C., Wei, X.-S., Liu, L. and Yang, J. (2017), Adversarial posenet: A structure-aware

convolutional network for human pose estimation, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 1212–1221.

Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y. (2014), 'On the properties of neural machine translation: Encoder-decoder approaches', *arXiv:1409.1259* .

Comaniciu, D. and Meer, P. (2002), 'Mean shift: a robust approach toward feature space analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5), 603–619.

Corman, E. and Ovsjanikov, M. (2019), 'Functional characterization of deformation fields', *ACM Transactions on Graphics* **38**(1), 8.

Corman, É., Ovsjanikov, M. and Chambolle, A. (2014), Supervised descriptor learning for non-rigid shape matching., *in* 'Proceedings of the European Conference on Computer Vision Workshops', pp. 283–298.

Corman, E., Solomon, J., Ben-Chen, M., Guibas, L. and Ovsjanikov, M. (2017), 'Functional characterization of intrinsic and extrinsic geometry', *ACM Transactions on Graphics* **36**(2), 14.

Cosmo, L., Rodola, E., Masci, J., Torsello, A. and Bronstein, M. M. (2016), Matching deformable objects in clutter, *in* 'Proceedings of the IEEE Conference on 3D Vision', pp. 1–10.

Cunningham, M. R., Roberts, A. R., Barbee, A. P., Druen, P. B. and Wu, C.-H. (1995), '"their ideas of beauty are, on the whole, the same as ours": Consistency and variability in the cross-cultural perception of female physical attractiveness', *Journal of Personality and Social Psychology* **68**(2), 261.

DeRose, T., Kass, M. and Truong, T. (1998), Subdivision surfaces in character animation, *in* 'Proceedings of the 25th annual conference on Computer graphics and interactive techniques', pp. 85–94.

Dokmanic, I., Parhizkar, R., Ranieri, J. and Vetterli, M. (2015), 'Euclidean distance matrices: Essential theory, algorithms, and applications', *IEEE Signal Processing Magazine* **32**(6), 12–30.

Dutagaci, H., Cheung, C. and Godil, A. (2012), 'Evaluation of 3d interest point detection techniques via human-generated ground truth', *The Visual Computer* **28**(9), 901–917.

Eisenthal, Y., Dror, G. and Ruppin, E. (2006), 'Facial attractiveness: Beauty and the machine', *Neural Computation* **18**(1), 119–142.

Farin, G. (2014), *Curves and surfaces for computer-aided geometric design: a practical guide*, Elsevier.

Farris, F. A. (2010), 'The gini index and measures of inequality', *American Mathematical Monthly* **117**(10), 851–864.

Feige, U., Peleg, D. and Kortsarz, G. (2001), 'The dense k-subgraph problem', *Algorithmica* **29**(3), 410–421.

Feixas, M., Sbert, M. and González, F. (2009), 'A unified information-theoretic framework for viewpoint selection and mesh saliency', *ACM Transactions on Applied Perception* **6**(1), 1–23.

Fu, K., Gu, I. Y.-H. and Yang, J. (2017), 'Saliency detection by fully learning a continuous conditional random field', *IEEE Transactions on Multimedia* **19**(7), 1531–1544.

Gal, R. and Cohen-Or, D. (2006), 'Salient geometric features for partial shape matching and similarity', *ACM Transactions on Graphics* **25**(1), 130–150.

Gan, J., Li, L., Zhai, Y. and Liu, Y. (2014), 'Deep self-taught learning for facial beauty prediction', *Neurocomputing* **144**, 295–303.

Garland, M. and Heckbert, P. S. (1997), Surface simplification using quadric error metrics, *in* 'Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques', ACM, pp. 209–216.

Gelfand, N., Mitra, N. J., Guibas, L. J. and Pottmann, H. (2005), Robust global registration, *in* 'Proceedings of the Eurographics Symposium on Geometry Processing', Eurographics Association.

Georgescu, B., Shimshoni, I. and Meer, P. (2003), Mean shift based clustering in high dimensions: a texture classification example, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 456–463.

Giorgi, D., Biasotti, S. and Paraboschi, L. (2007), 'Shape retrieval contest 2007: Watertight models track'.

Gkioxari, G., Malik, J. and Johnson, J. (2019), 'Mesh r-cnn', *arXiv preprint arXiv:1906.02739* .

Godsil, C. D., Royle, G. and Godsil, C. (2001), *Algebraic graph theory*, Springer New York.

Gokaslan, A., Ramanujan, V., Ritchie, D., In Kim, K. and Tompkin, J. (2018), Improving shape deformation in unsupervised image-to-image translation, *in* 'Proceedings of European Conference on Computer Vision', pp. 649–665.

Gooch, B., Reinhard, E., Moulding, C. and Shirley, P. (2001), Artistic composition for image creation, *in* 'Rendering Techniques', Springer Vienna, pp. 83–88.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014), Generative adversarial nets, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 2672–2680.

Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A. and Bengio, Y. (2013), Maxout networks, *in* 'Proceedings of the International Conference on Machine Learning', pp. 1319–1327.

Grammer, K. and Thornhill, R. (1994), 'Human facial attractiveness and sexual selection: The role of symmetry and averageness', *Journal of Comparative Psychology* **108**(3), 233.

Gu, M., Hu, S., Wang, X., Liang, X., Shen, X. and Qin, A. (2014), Saliency-driven depth compression for 3d image warping, *in* 'Proceedings of Pacific Graphics (short paper)', pp. 91–96.

Gu, X., Gortler, S. J. and Hoppe, H. (2002), Geometry images, *in* 'Proceedings of the 29th annual conference on Computer graphics and interactive techniques', pp. 355–361.

Guo, D. and Sim, T. (2009), Digital face makeup by example, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 73–79.

Habibie, I., Holden, D., Schwarz, J., Yearsley, J. and Komura, T. (2017), A recurrent variational autoencoder for human motion synthesis., *in* 'Proceedings of British Machine Vision Conference'.

Harel, J., Koch, C. and Perona, P. (2006), Graph-based visual saliency, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 545–552.

Hassner, T. (2013), Viewing real-world faces in 3d, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 3607–3614.

He, K., Zhang, X., Ren, S. and Sun, J. (2015), Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 1026–1034.

He, S., Lau, R. W. and Yang, Q. (2016), Exemplar-driven top-down saliency detection via deep association, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 5723–5732.

Hitomi, E. E., Silva, J. V. and Ruppert, G. C. (2015), 3d scanning using rgbd imaging devices: A survey, *in* 'Developments in Medical Image Processing and Computational Vision', pp. 379–395.

Hochreiter, S. and Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

Hoppe, H. (1996), Progressive meshes, *in* 'Proceedings of the 23rd annual conference on Computer graphics and interactive techniques', pp. 99–108.

Hou, X. and Zhang, L. (2007), Saliency detection: A spectral residual approach, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 1–8.

Howard, I. P. (2002), *Seeing in depth*, University of Toronto Press.

Howlett, S., Hamill, J. and O'Sullivan, C. (2005), 'Predicting and evaluating saliency for simplified polygonal models', *ACM Transactions on Applied Perception* **2**(3), 286–308.

Hu, S., Bhattacharya, H., Chattopadhyay, M., Aslam, N. and Shum, H. P. (2018), A dual-stream recurrent neural network for student feedback prediction using kinect, *in* 'Proceedings of the IEEE International Conference on Software, Knowledge, Information Management & Applications', pp. 1–8.

Hu, S., Rueangsirarak, W., Bouchée, M., Aslam, N. and Shum, H. P. (2017), A motion classification approach to fall detection, *in* 'Proceedings of the IEEE International Conference on Software, Knowledge, Information Management & Applications', pp. 1–6.

Hu, S., Shum, H. P. and Mucherino, A. (2019), Dspp: Deep shape and pose priors of humans,

in 'Proceedings of the ACM SIGGRAPH International Conference on Motion, Interaction, and Games (short paper)', pp. 1–6.

Hurley, N. and Rickard, S. (2009), 'Comparing measures of sparsity', *TIT* **55**(10), 4723–4741.

Ioffe, S. and Szegedy, C. (2015), Batch normalization: Accelerating deep network training by reducing internal covariate shift, *in* 'Proceedings of the International Conference on Machine Learning', pp. 448–456.

Itti, L., Koch, C. and Niebur, E. (1998), 'A model of saliency-based visual attention for rapid scene analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(11), 1254–1259.

Jeong, S.-W. and Sim, J.-Y. (2017), 'Saliency detection for 3d surface geometry using semi-regular meshes', *IEEE Transactions on Multimedia* **19**(12), 2692–2705.

Ji, Y., Zhang, H., Tseng, K.-K., Chow, T. W. and Wu, Q. J. (2019), 'Graph model-based salient object detection using objectness and multiple saliency cues', *Neurocomputing* **323**, 188–202.

Johnson, A. and Hebert, M. (1999), 'Using spin images for efficient object recognition in cluttered 3d scenes', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(5), 433–449.

Joshi, N., Matusik, W., Adelson, E. H. and Kriegman, D. J. (2010), 'Personal photo enhancement using example images', *ACM Transactions on Graphics* **29**(2), 12:1–12:15.

Kanazawa, A., Black, M. J., Jacobs, D. W. and Malik, J. (2018), End-to-end recovery of human shape and pose, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 7122–7131.

Kazhdan, M., Funkhouser, T. and Rusinkiewicz, S. (2003), Rotation invariant spherical harmonic representation of 3d shape descriptors, *in* 'Proceedings of the Eurographics Symposium on Geometry Processing', Eurographics Association, pp. 156–164.

Kemelmacher-Shlizerman, I., Shechtman, E., Garg, R. and Seitz, S. M. (2011), 'Exploring photobios', *ACM Transactions on Graphics* **30**(4), 61:1–61:10.

Kim, V. G., Lipman, Y. and Funkhouser, T. (2011), 'Blended intrinsic maps', *ACM Transactions on Graphics* **30**(4), 79.

Kim, Y. and Varshney, A. (2006), 'Saliency-guided enhancement for volume visualization', *IEEE Transactions on Visualization and Computer Graphics* **12**(5), 925–932.

Kim, Y. and Varshney, A. (2008), 'Persuading visual attention through geometry', *IEEE Transactions on Visualization and Computer Graphics* **14**(4), 772–782.

Kim, Y., Varshney, A., Jacobs, D. W. and Guimbretière, F. (2010), 'Mesh saliency and human eye fixations', *ACM Transactions on Applied Perception* **7**(2), 1–13.

Kingma, D. P. and Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980* .

Kingma, D. P. and Welling, M. (2013), 'Auto-encoding variational bayes', *arXiv preprint arXiv:1312.6114* .

Koch, C. and Ullman, S. (1987), Shifts in selective visual attention: Towards the underlying neural circuitry, *in* 'Matters of Intelligence', Vol. 188, Springer Netherlands, pp. 115–141.

Kortgen, M., Novotni, M. and Klein, R. (2003), 3d shape matching with 3d shape contexts, *in* 'Proceedings of the 7th Central European Seminar on Computer Graphics'.

Kraevoy, V. and Sheffer, A. (2004), 'Cross-parameterization and compatible remeshing of 3d models', *ACM Transactions on Graphics* **23**, 861–869.

Lassner, C., Pons-Moll, G. and Gehler, P. V. (2017), A generative model of people in clothing, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 853–862.

Laurentini, A. and Bottino, A. (2014), 'Computer analysis of face beauty: A survey', *Computer Vision and Image Understanding* **125**, 184–199.

Le, V., Brandt, J., Lin, Z., Bourdev, L. and Huang, T. S. (2012), Interactive facial feature localization, *in* 'Proceedings of the Euporean Conference on Computer Vision', pp. 679–692.

Lee, C. H., Varshney, A. and Jacobs, D. W. (2005), 'Mesh saliency', *ACM Transactions on Graphics* **24**(3), 659–666.

Leifman, G., Shtrom, E. and Tal, A. (2012), Surface regions of interest for viewpoint selection, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 414–421.

Leonard, M. J. (1997), An image-based approach to three-dimensional computer graphics, PhD thesis, University of North Carolina at Chapel Hill.

Leordeanu, M., Sukthankar, R. and Hebert, M. (2012), 'Unsupervised learning for graph matching', *International Journal of Computer Vision* **96**(1), 28–45.

Leyvand, T., Cohen-Or, D., Dror, G. and Lischinski, D. (2008), 'Data-driven enhancement of facial attractiveness', *ACM Transactions on Graphics* **27**(3), 38:1–38:9.

Li, J., Xiong, C., Liu, L., Shu, X. and Yan, S. (2015), Deep face beautification, *in* 'Proceedings of the ACM International Conference on Multimedia', pp. 793–794.

Li, Y., Hou, X., Koch, C., Rehg, J. M. and Yuille, A. (2014), The secrets of salient object segmentation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition'.

Liao, J., Lima, R. S., Nehab, D., Hoppe, H., Sander, P. V. and Yu, J. (2014), 'Automating image morphing using structural similarity on a halfway domain', *ACM Transactions on Graphics* **33**(5), 168:1–168:12.

Liao, Q., Jin, X. and Zeng, W. (2012), 'Enhancing the symmetry and proportion of 3d face geometry', *IEEE Transactions on Visualization and Computer Graphics* **18**(10), 1704–1716.

Lipman, Y. and Funkhouser, T. (2009), 'Mobius voting for surface correspondence', *ACM Transactions on Graphics* **28**(3), 72:1–72:12.

Litany, O., Remez, T., Rodola, E., Bronstein, A. M. and Bronstein, M. M. (2017), Deep functional maps: Structured prediction for dense shape correspondence, *in* 'Proceedings of the IEEE International Conference on Computer Vision', Vol. 2, p. 8.

Litman, R. and Bronstein, A. M. (2014), 'Learning spectral descriptors for deformable shape correspondence', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(1), 171–180.

Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X. and Shum, H.-Y. (2011), 'Learning to detect a salient object', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(2), 353–367.

Liu, Y.-S., Liu, M., Kihara, D. and Ramani, K. (2007), Salient critical points for meshes, *in* 'Proceedings of the ACM Symposium on Solid and Physical Modeling', ACM, pp. 277–282.

Longhurst, P., Debattista, K. and Chalmers, A. (2006), A gpu based saliency map for high-fidelity selective rendering, *in* 'Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa', ACM, pp. 21–29.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G. and Black, M. J. (2015), 'Smpl: A skinned multi-person linear model', *ACM Transactions on Graphics* **34**(6), 248.

Luebke, D. and Hallen, B. (2001), Perceptually driven simplification for interactive rendering, *in* 'Rendering Techniques', Springer Vienna, pp. 223–234.

Luebke, D., Watson, B., Cohen, J. D., Reddy, M. and Varshney, A. (2002), *Level of Detail for 3D Graphics*, Elsevier Science Inc.

Magnus, J. R. (1985), 'On differentiating eigenvalues and eigenvectors', *Econometric Theory* **1**(02), 179–191.

Maleš, L., Marčetić, D. and Ribarić, S. (2019), 'A multi-agent dynamic system for robust multi-face tracking', *Expert Systems with Applications* **126**, 246–264.

Mantiuk, R., Myszkowski, K. and Pattanaik, S. (2003), Attention guided mpeg compression for computer animations, *in* 'Proceedings of the 19th Spring Conference on Computer Graphics', ACM, pp. 239–244.

Maron, H., Dym, N., Kezurer, I., Kovalsky, S. and Lipman, Y. (2016), 'Point registration via efficient convex relaxation', *ACM Transactions on Graphics* **35**(4), 73.

Masci, J., Boscaini, D., Bronstein, M. and Vandergheynst, P. (2015), Geodesic convolutional neural networks on riemannian manifolds, *in* 'Proceedings of the IEEE International Conference on Computer Vision workshops', pp. 37–45.

McDonnell, R., Larkin, M., Hernández, B., Rudomin, I. and O'Sullivan, C. (2009), 'Eye-catching crowds: Saliency based selective variation', *ACM Transactions on Graphics* **28**(3), 1–10.

Meyer, M., Desbrun, M., Schroder, P. and Barr, A. H. (2002), 'Discrete differential-geometry operators for triangulated 2-manifolds', *Mathematics & Visualization* **6**(8-9), 35–57.

Meyer, M., Desbrun, M., Schröder, P. and Barr, A. H. (2003), Discrete differential-geometry operators for triangulated 2-manifolds, *in* 'Visualization and mathematics III', Springer, pp. 35–57.

Miao, Y. and Feng, J. (2010), 'Perceptual-saliency extremum lines for 3d shape illustration', *The Visual Computer* **26**(6-8), 433–443.

Miyato, T., Kataoka, T., Koyama, M. and Yoshida, Y. (2018), 'Spectral normalization for generative adversarial networks', *arXiv preprint arXiv:1802.05957* .

Nair, V. and Hinton, G. E. (2010), Rectified linear units improve restricted boltzmann machines, *in* 'Proceedings of International Conference on Machine Learning', pp. 807–814.

Nealen, A., Igarashi, T., Sorkine, O. and Alexa, M. (2006), Laplacian mesh optimization, *in* 'Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia', pp. 381–389.

Novatnack, J. and Nishino, K. (2007), Scale-dependent 3d geometric features, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 1–8.

Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A. and Guibas, L. (2012), 'Functional maps: a flexible representation of maps between shapes', *ACM Transactions on Graphics* **31**(4), 30.

Ovsjanikov, M., Mérigot, Q., Mémoli, F. and Guibas, L. (2010), 'One point isometric matching with the heat kernel', *Computer Graphics Forum* **29**(5), 1555–1564.

Pajak, D., Herzog, R., Eisemann, E., Myszkowski, K. and Seidel, H.-P. (2011), 'Scalable remote rendering with depth and motion-flow augmented streaming.', *Computer Graphics Forum* **30**(2), 415–424.

Pasupa, K., Sunhem, W. and Loo, C. K. (2019), 'A hybrid approach to building face shape classifier for hairstyle recommender system', *Expert Systems with Applications* **120**, 14–32.

Pingping, T., Junjie, C., Shuhua, L., Xiuping, L. and Ligang, L. (2015), 'Mesh saliency via ranking unsalient patches in a descriptor space', *Computers & Graphics* **46**, 264–274.

Pishchulin, L., Wuhrer, S., Helten, T., Theobalt, C. and Schiele, B. (2017), 'Building statistical shape spaces for 3d human modeling', *Pattern Recognition* **67**, 276–286.

Qi, C. R., Su, H., Mo, K. and Guibas, L. J. (2017*a*), Pointnet: Deep learning on point sets for 3d classification and segmentation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition'.

Qi, C. R., Su, H., Mo, K. and Guibas, L. J. (2017*b*), Pointnet: Deep learning on point sets for 3d classification and segmentation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 652–660.

Qi, C. R., Yi, L., Su, H. and Guibas, L. J. (2017), Pointnet++: Deep hierarchical feature learning on point sets in a metric space, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 5099–5108.

Qu, L. and Meyer, G. W. (2008), 'Perceptually guided polygon reduction', *IEEE Transactions on Visualization and Computer Graphics* **14**(5), 1015–1029.

Rickard, S. and Hurley, N. (2008), 'Comparing measures of sparsity', *IEEE Transactions on Information Theory* **55**(10), 55–60.

Rodolà, E., Rota Bulo, S., Windheuser, T., Vestner, M. and Cremers, D. (2014), Dense non-rigid shape correspondence using random forests, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 4177–4184.

Rustamov, R. M. (2007), Laplace-beltrami eigenfunctions for deformation invariant shape representation, *in* 'Proceedings of the The 5th Eurographics Symposium on Geometry Processing', pp. 225–233.

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S. and Pantic, M. (2013), A semi-automatic methodology for facial landmark annotation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 896–903.

Sahillioğlu, Y. and Yemez, Y. (2011), 'Coarse-to-fine combinatorial matching for dense isometric shape correspondence', *Computer Graphics Forum* **30**(5), 1461–1470.

Salti, S., Tombari, F. and Di Stefano, L. (2014), 'Shot: Unique signatures of histograms for surface and texture description', *Computer Vision and Image Understanding* **125**, 251–264.

Schaefer, S., McPhail, T. and Warren, J. (2006), 'Image deformation using moving least squares', *ACM Transactions on Graphics* **25**(3), 533–540.

Scherbaum, K., Ritschel, T., Hullin, M., Thormählen, T., Blanz, V. and Seidel, H.-P. (2011), 'Computer-suggested facial makeup', **30**(2), 485–492.

Schmid, K., Marx, D. and Samal, A. (2008), 'Computation of a face attractiveness index based on neoclassical canons, symmetry, and golden ratios', *Pattern Recognition* **41**(8), 2710–2717.

Schneider, P. and Eberly, D. H. (2002), *Geometric tools for computer graphics*, Elsevier.

Secord, A., Lu, J., Finkelstein, A., Singh, M. and Nealen, A. (2011), 'Perceptual models of viewpoint preference', *ACM Transactions on Graphics* **30**(5), 1–12.

Seidman, G. and Miller, O. S. (2013), 'Effects of gender and physical attractiveness on visual attention to facebook profiles', *Cyberpsychology, Behavior, and Social Networking* **16**(1), 20–24.

SFU Motion Capture dataset (n.d.), http://mocap.cs.sfu.ca/.

Sheffer, A., Gotsman, C. and Dyn, N. (2004), 'Robust spherical parameterization of triangular meshes', *Computing* **72**(1-2), 185–193.

Shilane, P. and Funkhouser, T. (2007), 'Distinctive regions of 3d surfaces', *ACM Transactions on Graphics* **26**(2).

Shilane, P., Min, P., Kazhdan, M. and Funkhouser, T. (2004), The princeton shape benchmark, *in* 'Proceedings Shape Modeling Applications', pp. 167–178.

SHREC'2007 watertight mesh database (n.d.), http://watertight.ge.imati.cnr.it/.

Sinha, A., Bai, J. and Ramani, K. (2016), Deep learning 3d shape surfaces using geometry images, *in* 'Proceedings of the European Conference on Computer Vision', pp. 223–240.

Sipiran, I. and Bustos, B. (2010), A robust 3d interest points detector based on harris operator, *in* 'Proceedings of the 3rd Eurographics Conference on 3D Object Retrieval', Eurographics Association, pp. 7–14.

Slater, A., Von der Schulenburg, C., Brown, E., Badenoch, M., Butterworth, G., Parsons, S. and Samuels, C. (1998), 'Newborn infants prefer attractive faces', *Infant Behavior and Development* **21**(2), 345–354.

Song, R., Liu, Y., Martin, R. R. and Rosin, P. L. (2014), 'Mesh saliency via spectral processing', *ACM Transactions on Graphics* **33**(1), 1–17.

Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C. and Seidel, H.-P. (2004), Laplacian surface editing, *in* 'Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing', pp. 175–184.

Sun, J., Ovsjanikov, M. and Guibas, L. (2009), 'A concise and provably informative multi-scale signature based on heat diffusion', *Computer Graphics Forum* **28**(5), 1383–1392.

Sundstedt, V., Gutierrez, D., Anson, O., Banterle, F. and Chalmers, A. (2007), 'Perceptual rendering of participating media', *ACM Transactions on Applied Perception* **4**(3), 15.

Tan, Q., Gao, L., Lai, Y.-K. and Xia, S. (2018), Variational autoencoders for deforming 3d mesh models, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 5841–5850.

Tarini, M., Hormann, K., Cignoni, P. and Montani, C. (2004), 'Polycube-maps', *ACM Transactions on Graphics* **23**(3), 853–860.

Tena, J. R., De la Torre, F. and Matthews, I. (2011), 'Interactive region-based linear 3d face models', *ACM Transactions on Graphics* **30**(4), 76:1–76:10.

The AIM shape repository (n.d.), http://shapes.aimatshape.net/.

The Stanford 3D scanning repository (n.d.), http://graphics.stanford.edu/data/3Dscanrep/.

Tombari, F., Salti, S. and Di Stefano, L. (2010), Unique shape context for 3d data description, *in* 'Proceedings of the ACM workshop on 3D object retrieval', ACM, pp. 57–62.

Treisman, A. M. and Gelade, G. (1980), 'A feature-integration theory of attention', *Cognitive Psychology* **12**(1), 97 – 136.

Turk, G. (1992), Re-tiling polygonal surfaces, *in* 'Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques', ACM, pp. 55–64.

Vallet, B. and Lévy, B. (2008), Spectral geometry processing with manifold harmonics, *in* 'Computer Graphics Forum', Vol. 27, Wiley Online Library, pp. 251–260.

Van Kaick, O., Zhang, H., Hamarneh, G. and Cohen-Or, D. (2011), 'A survey on shape correspondence', *Computer Graphics Forum* **30**(6), 1681–1707.

Vázquez, P.-P., Feixas, M., Sbert, M. and Heidrich, W. (2001), Viewpoint selection using viewpoint entropy., *in* 'Proceedings of the International Symposium on Vision, Modeling, and Visualization', Vol. 1, pp. 273–280.

Vestner, M., Litman, R., Rodolà, E., Bronstein, A. and Cremers, D. (2017), Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition'.

Vlasic, D., Baran, I., Matusik, W. and Popović, J. (2008), 'Articulated mesh animation from multi-view silhouettes', *ACM Transactions on Graphics* **27**(3), 97.

Voigt, P. and Von dem Bussche, A. (2017), 'The eu general data protection regulation (gdpr)', *A Practical Guide, 1st Ed., Cham: Springer International Publishing* .

Von Luxburg, U. (2007), 'A tutorial on spectral clustering', *Statistics and computing* **17**(4), 395–416.

Wang, J., Cheng, Y. and Schmidt Feris, R. (2016), Walk and learn: Facial attribute representation learning from egocentric video and contextual data, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 2295–2304.

Wang, Y., Jacobson, A., Barbič, J. and Kavan, L. (2015), 'Linear subspace design for real-time shape deformation', *ACM Transactions on Graphics* **34**(4), 57.

Wei, L., Huang, Q., Ceylan, D., Vouga, E. and Li, H. (2016), Dense human body correspondences using convolutional networks, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 1544–1553.

Winston, J. S., O'Doherty, J., Kilner, J. M., Perrett, D. I. and Dolan, R. J. (2007), 'Brain systems for assessing facial attractiveness', *Neuropsychologia* **45**(1), 195–206.

Wu, J., Shen, X., Zhu, W. and Liu, L. (2013), 'Mesh saliency with global rarity', *Graphical Models* **75**(5), 255 – 264.

Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, B. and Tenenbaum, J. (2017), Marrnet: 3d shape reconstruction via 2.5 d sketches, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 540–550.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J. (2015), 3d shapenets: A deep representation for volumetric shapes, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1912–1920.

Xia, S., Gao, L., Lai, Y.-K., Yuan, M.-Z. and Chai, J. (2017), 'A survey on human performance capture and animation', *Journal of Computer Science and Technology* **32**(3), 536–554.

Xiao, H., Feng, J., Wei, Y., Zhang, M. and Yan, S. (2018), 'Deep salient object detection with dense connections and distraction diagnosis', *IEEE Transactions on Multimedia* .

Yamauchi, H., Saleem, W., Yoshizawa, S., Karni, Z., Belyaev, A. and Seidel, H. P. (2006), Towards stable and salient multi-view representation of 3d shapes, *in* 'Proceedings of the IEEE International Conference on Shape Modeling and Applications', pp. 40–40.

Yang, F., Bourdev, L., Shechtman, E., Wang, J. and Metaxas, D. (2012), Facial expression editing in video using a temporally-smooth factorization, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 861–868.

Yang, F., Wang, J., Shechtman, E., Bourdev, L. and Metaxas, D. (2011), 'Expression flow for 3d-aware face component transfer', *ACM Transactions on Graphics* **30**(4), 60:1–60:10.

Yee, H., Pattanaik, S. and Greenberg, D. P. (2001), 'Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments', *ACM Transactions on Graphics* **20**(1), 39–65.

Yi, L., Su, H., Guo, X. and Guibas, L. (2017), Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition'.

Yuan, X.-T. and Zhang, T. (2013), 'Truncated power method for sparse eigenvalue problems', *Journal of Machine Learning Research* **14**(1), 899–925.

Zhang, D., Zhao, Q. and Chen, F. (2011), 'Quantitative analysis of human facial beauty using geometric features', *Pattern Recognition* **44**(4), 940–950.

Zhang, E., Mischaikow, K. and Turk, G. (2005), 'Feature-based surface parameterization and texture mapping', *ACM Transactions on Graphics* **24**(1), 1–27.

Zhang, H., Van Kaick, O. and Dyer, R. (2010), Spectral mesh processing, *in* 'Computer graphics forum', Vol. 29, Wiley Online Library, pp. 1865–1894.

Zhang, J., Shan, S., Kan, M. and Chen, X. (2014), Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment, *in* 'Proceedings of the European Conference on Computer Vision', pp. 1–16.

Zhang, L., Dong, H. and Saddik, A. E. (2016), 'From 3d sensing to printing: A survey', *ACM Transactions on Multimedia Computing, Communications, and Applications* **12**(2), 27.

Zhang, L., Shum, H. P. H., Liu, L., Guo, G. and Shao, L. (2019), 'Multiview discriminative marginal metric learning for makeup face verification', *Neurocomputing* **333**, 339–350.

Zhang, L., Tong, M. H., Marks, T. K., Shan, H. and Cottrell, G. W. (2008), 'Sun: A bayesian framework for saliency using natural statistics', *Journal of Vision* **8**(7).

Zhang, L., Zhang, D., Sun, M.-M. and Chen, F.-M. (2017), 'Facial beauty analysis based on geometric feature: Toward attractiveness assessment application', *Expert Systems with Applications* **82**, 252–265.

Zhang, Z. (2012), 'Microsoft kinect sensor and its effect', *IEEE multimedia* **19**(2), 4–10.

Zhao, Q. (2009), 'A survey on virtual reality', *Science in China Series F: Information Sciences* **52**(3), 348–400.

Zhao, X., Wang, H. and Komura, T. (2014), 'Indexing 3d scenes using the interaction bisector surface', *ACM Transactions on Graphics* **33**(3), 22.

Zhao, Y., Liu, Y., Wang, Y., Wei, B., Yang, J., Zhao, Y. and Wang, Y. (2016), 'Region-based saliency estimation for 3d shape analysis and understanding', *Neurocomputing* **197**, 1–13.

Zhou, D., Weston, J., Gretton, A., Bousquet, O. and Scholkopf, B. (2003), Ranking on data manifolds, *in* 'Proceedings of Advances in Neural Information Processing Systems', pp. 169–176.

Zyda, M. (2005), ‘From visual simulation to virtual reality to games’, *Computer* **38**(9), 25–32.