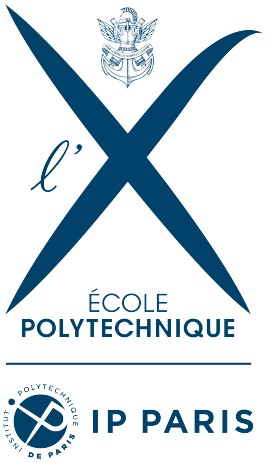


ECOLE POLYTECHNIQUE

Master's Thesis in Artificial Intelligence and Advanced Visual Computing

Implicit latent space exploration for Shape Optimisation and Correspondence

Author: RamanaSubramanyam Sundararaman
Supervisor: Professor Maks Ovsjanikov
Submission Date: 15/09/2021

Abstract

Shape optimisation and matching are two fundamental problems in 3D computer vision with far-reaching scope of application areas such as industrial design engineering, 3D content generation, medical imaging, etc. While the vast majority of literature in 3D shape optimisation are geared towards single use-cases, often the real-world applications entail a more generic optimisation pipeline which can successfully handle multiple constraints simultaneously. To this end, in this thesis, we propose simple yet comprehensive techniques for optimising different physical properties of shape represented as a latent vector. Particularly in settings where limited training data is a key concern, we demonstrate that our method is robust and produces geometrically plausible shapes while respecting the constraints. Leveraging the strong expressive power of Deep Implicit 3D models, we show that such latent representations are not only suitable for shape optimisation but for non-rigid shape matching such as articulated humans. Unlike classical correspondence techniques, based on triangle mesh or point-cloud representations, the resulting network is remarkably robust in the presence of strong artefacts, including significant noise and missing data.

Contents

1. Introduction

3D Shape Optimisation, which typically refers to the task of generating objects that are structurally plausible and visually innovative is a central problem in Computer Vision, Graphics and Machine Learning. This has a wide ranging use case, namely design, healthcare, entertainment, construction sectors to name a few. In particular, examples include but are not limited to generating user specific prosthetics, creative household appliances, raw material optimised CAD models for additive manufacturing etc. Standard practices in use today for achieving these tasks include significant expertise in artistic skills, 3D sculpting, meshing and UV Layout, involving significant time and expense for human labour.

On the other hand, with the advent of widely available annotated 3D benchmarks, there is a sharp increase in the use of generative models in AI assisted 3D design. This significant advancement has been made possible due to seminal works in deep probabilistic models such as Variational Autoencoder (VAE), Generative Adversarial Network (GANs), Variational Auto Decoder (VAD), etc. While such methods have resulted in an uncanny valley [1] for images and videos which are 2D representations of visual data, they fall short of producing such levels of realism in the 3D domain. This drawback can be attributed towards ambiguity in representation of 3D data, in contrast to unequivocal representation of 2D visual data as grids (or matrices) containing pixels. 3D data however can be represented as voxels, Point Clouds, Tri-mesh, Quad-mesh, RGB-D and multi-view images. Each representation has its own advantage and there exists no standard representation protocol. Thus unique data-driven approaches have to be modelled for each data representation, making this problem a widely open area of research.

Despite the aforementioned challenges, in the recent past a pioneering work in 3D modelling, DeepSDF [1] proposes to model shapes as Signed Distance Fields, (SDFs) through simple Multi-Layer Perceptrons (MLPs), making them agnostic to resolution, connectivity and input size. A key observation here is that a compact, lower dimensional latent representation learnt from data serves as a crucial factor to encode geometric information of objects in the neurons of the MLP. While the majority of recent research has been attempting to improve the visual quality of generated outputs, there has been a steady increase in the use of generative models for design oriented 3D modelling with real world constraints. This includes (1) Resolution agnostic shape synthesis [1], (2) Reasoning between shapes using parts [2, 3], (3) Shape recovery [4] and (4) Generating physically plausible objects [5]. Encouraging results

[1] https://en.wikipedia.org/wiki/Deepfake

demonstrated by the aforementioned methods have paved the way for us to further explore the prowess of implicit representation of 3D data. To this end, continuing the recent research frontiers in fully understanding the capabilities of implicit 3D generative models for problems fundamental to 3D Shape Analysis is the fundamental and overall goal of this dissertation.

1.1. Shape Optimisation

The goal of shape optimisation is to generate a 3D model, satisfying various physical constraints which leads to better structural stability, rotational dynamics, durability etc yet preserving geometric and visual similarity to an object of interest (or query). Changes in shape that can be enforced at an individual mesh level by manual intervention, such as bringing about a change in dimensions (length, breadth and width), surface area, volume etc are termed as extrinsic property optimisation. On the other hand, modifying properties such as mass, static friction, etc are termed as extrinsic property optimisation. To bring about the change in the former is mostly incidental in the sense one would not need a generative model to accomplish this task. The latter, however, is an intricate task to both humans as well as existing research literature as there is a strong reliance on feedback mechanisms from simulation software. Under such circumstances, having a good initial guess on the geometry through a generative model can help ameliorate the laborious human efforts involved in such tasks. To this end, we aim to build novel techniques that can both generate and optimise shapes while respecting different properties.

1.2. Non-Rigid Shape Matching

Given two non-rigid deformable objects, a source and a target, represented as a mesh, the task of non-rigid shape correspondence involves establishing a map between each point on the source to its corresponding point on the target. The definition of non-rigidness can be extended to any object pertaining to living things such as soft tissues, muscle fibers, organs etc. Not so surprisingly, even moderately elastic physical objects such as fabric or rubber can manifest such non-rigid deformation when subjected to external forces. This is a long standing open problem, central to many research fields, namely, Computer Vision, Graphics, Robotics, Drug Discovery, etc.

Existing state-of-the-art methods for non-rigid shape matching are either spectral methods which rely only on intrinsic information [6, 7, 8] or use point-wise extrinsic information [9, 10]. Intrinsic methods are guaranteed to be optimal under isometry and are highly effective even with slight perturbation to this assumption. However, such methods suffer from strong assumptions on connectivity and their performance deteriorates by simply re-meshing two given objects. Our key observation here is that generative models such as DeepSDF can

potentially be inert towards such connectivity dependence as they operate on an implicitly defined volume. In this work, we will demonstrate how a Deep Implicit Neural network geared towards shape generation task can be successfully applied to the task of non-rigid shape correspondence.

1.3. Broader impact

Our work in Shape Optimisation is broadly motivated by the hope of advancing AI assisted computer aided design. In particular, this project is aimed towards optimising the "Topload" property of bottles while constraining the volume while generating shapes that are similar to the given (query) shape. Topload is a property of an object obtained by simulation of geometry (typically triangular meshes) along with a query weight. The query weight here denotes the amount of plastic used for synthesising bottles. By finding an optimal design that is geometrically closer to a given query shape and has maximum permissible Topload while preserving the volume and amount of plastic used, one could generate novel bottle designs that consume less plastic and are more durable for any given volume. While the amount of weight optimised per shape is an insignificant amount, when considering large scale manufacturing, one can save significant plastic thus contributing to a greener society. We thank Danone, first for taking such an initiative and second for providing us with annotated data to explore this problem of constrained space Shape Optimisation.

1.4. Our goals and contributions

Ability to optimise one or more properties of a 3D object in a differentiable yet computationally feasible manner is an important open problem in the area of 3D Shape Analysis. However, existing techniques for shape optimisation require significant compute power, training data and are built to be task specific. To this end, our goal in this report will be to develop an optimisation pipeline which is robust and can handle multiple constraints without excessive computation cost. In order to optimise a shape, it is necessary to have a proficient generative model which can efficiently represent input in a lower dimensional latent space. We therefore perform an extensive preliminary study in order to identify a generative model that is robust and efficient in representing 3D shapes with high fidelity. Then, we propose novel methods for shape optimisation, wherein, we optimise multiple properties of the shape from its lower dimensional latent representation - which we refer to as Latent Space Optimisation (LSO). We further introspect into our solution and propose different optimisation techniques to improve robustness in the presence of limited training data. Our key observation here is that a well-learned latent representation can be optimised to reflect desired change in the shape geometry. Building upon this observation, we also demonstrate that our implicit latent shape

representation can be used for template deformation based non-rigid shape correspondence and show that our approach is more robust towards noise than other state-of-the-art existing approaches. In summary,

1. We perform a preliminary study where we extensively benchmark different reconstruction techniques to identify the most relevant one for our use-case.
2. We propose a novel technique for implicit Latent space Shape Optimisation (LSO) and build several improvements to the LSO to be robust in the presence of limited training data.
3. We demonstrate that our implicit latent vectors can be adapted to template based non-rigid shape correspondence and are remarkably robust in the presence of strong artefacts, including significant noise and missing data.

2. Preliminary Study

Our main objective in this chapter is twofold. First, we benchmark different implicit 3D reconstruction methods by comparing them on a canonical test set from an existing benchmark. Then, we explore novel ways of enhancing the reconstruction with a specific focus of reconstructing mesh level textures or high-frequency components by borrowing inspiration from signal processing literature. We demonstrate a significant enhancement in reconstruction accuracy stemming from our proposed enhancement on two dataset with naturally occurring mesh level textures.

2.1. Introduction

A good reconstruction approach is crucial for our end goal, a superior shape optimisation and shape correspondence through template deformation pipeline. To this end, in this chapter, we focus on discerning reconstruction methods which are robust towards noises, being able to generalise across all training shapes, provide a compact representation of shapes in lower dimension as latent vectors and have a continuous well-defined shape latent space. Furthermore, we expect the method to be able to reconstruct high-frequency components occurring as mesh-level textures in the 3D object, for example accurately representing non-rigidly deforming parts.

Despite the ubiquitous requirement for such methods, unfortunately, no perfect solution exists for this task that is guaranteed to be the best of all worlds. While methods for large-scale 3D reconstruction are capable of producing a continuous shape latent space and can generalise to training set shapes, they do not encode high-frequency components of the input signal (add footnote). Hence, we assimilate recent advancements in Signal Processing and Computational Imaging to propose a reconstruction method that satisfies all the desired properties to serve as a useful tool for Shape Optimisation.

2.2. Related work

In this section, we briefly survey the current literature in 3D shape modelling. The task of 3D Shape modelling can be broadly classified into two main categories - a) Explicit representation and b) Implicit representation. The former refers to approaches that models the precise

location of object vertices in a 3D space while the latter models how far a given point in the space is from the surface of the object, i.e its Signed Distance Function (SDF).

2.2.1. Explicit Shape Modelling

Data-driven methods for explicit 3D modelling can be categorised into three main methods based on their input data, namely point-based, mesh-based and voxel based approaches.

Point Cloud modelling

The goal of Point Cloud modelling is to generate unordered point-sets that resemble real-world 3D objects *without* connectivity information. Most notable methods for point-based 3D-Modelling are [11, 12, 13]. Achlioptas *et al.* [11] propose a deep Auto-Encoder that learns compact latent representations for Point Clouds. In addition, they also propose L-GAN (Latent-GAN) which is a generative model that produces novel latent vectors by training a GAN [14, 15] on the latent vectors pre-computed by the auto-encoder. In the same spirit, Yang*et al.* [13] proposes PointFlow which learns two-layered hierarchical distributions, namely that of plausible shapes and distribution of points using Normalizing Flow [16]. Rempe *et al.* [17] extend this hierarchical distribution to spatio-temporal Point Cloud generation using Ordinary Differential Equation (ODE) at the latent space. While these approaches produce high quality reconstruction, point clouds in themselves do not contain any topological information making them infeasible for modelling surfaces.

3D Voxel modelling

Voxels are non-parametric extension of 2D image grid to a 3D volume. While the extension itself appeals trivial, it paves way for adapting data-driven approaches originally built for 2D convolution on images on the 3D voxels [18, 19, 20]. However, the cubically growing storage size and computational complexity restricts the input resolution thereby limiting them for generating high-resolution (HR) reconstructions. To ameliorate this issue, a more efficient data structure, Octree has been explored as an alternative [21, 22, 23]. While this helps processing of voxels upto the resolution of 512^3, yet they fall short in producing high quality shape reconstructions, particularly owing to non-smooth normals.

Mesh Modelling

Triangular mesh (Tri-Mesh) is one of the most widely used representation of 3D objects, which is a piecewise planar approximation of a smooth surface, compact and efficient for rendering. Data-driven approaches for Tri-Mesh modelling can broadly be categorised into 1) Graph based approaches [24, 25, 26, 27] and 2) Part based approaches [28, 29, 30]. The

former processes meshes as graphs and employs parametrised learnable operations pertinent to graphs [31, 32] to learn a higher-dimensional latent representation of the meshes. While these methods show high-fidelity reconstruction results, their generalisation capability is quite low, germane to objects with low variations such as faces. On the other hand, the part based approaches attempt to bridge this gap by processing parts of more sophisticated individually. However, such approaches are highly sensitive to topological noises.

2.2.2. Implicit Shape Representation

Deep Implicit Neural Networks are emerging methods for efficient, differentiable and high-fidelity shape representation. The implicit representation of shapes, typically Signed Distance field are encoded in the shape latent vectors and neurons of neural networks, typically is a shallow Multi-Layer Perceptron (MLP). Their working principle is very simple yet strikingly effective. In its most generic form [1], this network takes points in 3D space as input and predicts the signed distance of the point with respect to the surface of the object. Once the network is trained, underlying implicit surfaces (SDF=0) can be rendered through classic surface extraction algorithms such as Marching Cubes [33].

This contemporary data-driven modelling of 3D objects has several compelling advantages in comparison to explicit methods, namely

- Implicit representations are resolution agnostic.
- Implicit representations can model any arbitrary topologies in comparison to point clouds and in mesh (and voxel) based methods where the notion of topology is absent and are fixed respectively.
- The learning process can be facilitated through shallow MLPs (typically 6 layered) in contrast to using deeper and more memory intensive networks.
- Implicit representations can learn continuous functions defined in a volume rather than learning points or meshes which are discrete representations of data.
- Using an implicit representation, 3D shapes can be efficiently represented as vector in a higher dimension.

Given our end goal of shape optimisation and shape correspondence, we use the implicit representation, where, the 3D shapes are learnt in a differentiable manner. While the field of Deep Implicit Shape modelling is expansive, we restrict our focus in this section on approaches that are relevant to our final goal. More specifically, we briefly review different implicit shape modelling methods which we have used in this report for benchmarking.

DeepSDF

The main tool that we will use for reconstruction is DeepSDF [1], a simple yet powerful method for modelling 3D shape as continuous volumetric field. Each point within that volume is associated to a scalar value representing the point's distance from the surface, with the sign denoting whether or not the point is inside or outside the surface of interest. An exemplar of such implicit representation in comparison to standard Triangular mesh is shown in Figure 2.1. Denoting the implicit neural network with parameters θ as f_θ, and $p \in \mathbb{R}^3$ be a point near the surface of shape indexed by i whose latent representation is α_i. Then, the objective of this network is to predict the distance of p from the surface of object,

$$f_\theta\left(\vec{\alpha}_i, p\right) \approx SDF^i(\boldsymbol{p}) \tag{2.1}$$

At training time, the following loss is minimised, which penalises deviation in predicted signed distance and minimise the norm of latent vectors, z_i to promote compactness.

$$\arg\min_{\theta,\{\vec{\alpha}_i\}_{i=1}^N} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} \left| f_\theta(p) - s \right| + \frac{1}{\sigma^2} \left\| \vec{\alpha}_i \right\|_2^2 \right) \tag{2.2}$$

At the test time, θ is fixed while optimal latent code $\vec{\alpha}_i$ is estimated by Maximum-A-Posterior (MAP) as

$$\hat{\vec{\alpha}} = \arg\min_{\vec{\alpha}} \sum_{(p_j, s_j) \in X} \left| f_\theta\left(\vec{\alpha}, p_j\right) - s_j \right| + \frac{1}{\sigma^2} \|\vec{\alpha}\|_2^2 \tag{2.3}$$

DualSDF

DualSDF [3] is another Deep Implicit method which builds on DeepSDF, but has two important additional functionalities. First, it represent shapes as a combination of a fixed number of primitives - unit spheres, where the SDF of a point in shape space is approximated as the SDF to the closest unit sphere. Second, they use a Variational Autodecoder (VAD) framework [34] instead of Autodecoder (AD) framework used by DeepSDF [1]. In the former latent vectors are modelled as parameters of approximate posterior distribution, which is a Gaussian with diagonal covariance. At training time, the distribution of latent vectors are encouraged to be closer to the original shape distribution by minimising the divergence between the two probability distributions (or maximising ELBO), promoting compactness of the latent space.

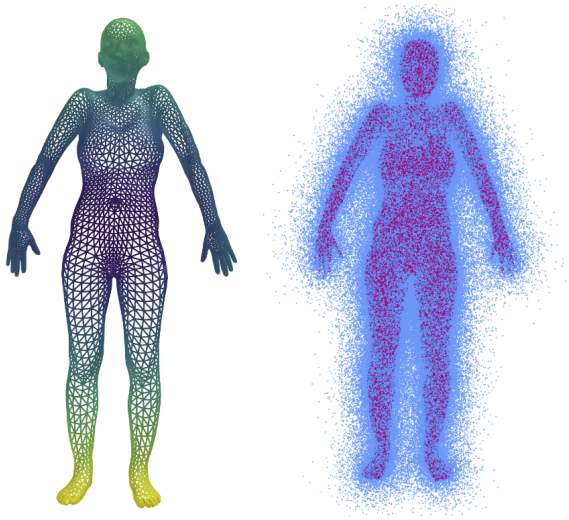


Figure 2.1.: Standard Tri-Mesh representation of 3D object (left) and Signed Distance Field representation of the same object (right). Blue denotes points in space that are *outside* the surface and red denotes points in space *inside* the surface.

IM-Net

IM-Net [35] is a parallel work to DeepSDF which proposes to learn continuous SDF within a volume as a function of parametrised MLPs. Noteworthy difference with DeepSDF is that IM-Net applies series of 3D convolution over voxel grids to create a latent representation in contrast to an auto-decoder based approach. Then, original point coordinates of the mesh are concatenated to this latent vector which is then passed onto a parametrised MLPs to predict the whether the point is inside or outside the object.

PQ-Net

PQ-Net [2] is a part-based approach, wherein, each part of a shape is segmented and encoded into a feature vector using a seq2seq part encoder [36]. The input to the auto encoder consists of a geometric feature vector, bounding box coordinates, the translation and scaling factors of the local frame according to a global coordinate system. Then, each encoded parts are decoded using GRUs [37], which decode one part at a time. This decoder at test time can predict arbitrary number of parts, which are encouraged to be consistent using a novel stop loss.

2.3. Benchmarking

In this section, we compare different implicit shape representation approaches discussed above. In a broad sense, our comparison is aimed towards finding the optimal method between Auto-Encoder vs Encoder-free and Part-based vs global approaches. We demonstrate our comparison on Chairs from ShapeNetV2 dataset [38]. The reason for our choice is that chairs are topologically simple object, has relatively low high-frequency details, distinct parts and whose surface can undergo only a small degree of non-rigid deformation.

For DeepSDF and DualSDF, we train the model from scratch while for IM-Net and PQ-Net, we use the pre-trained model provided by the author. DeepSDF was trained for 2000 epochs while DualSDF was trained for 2800 epochs. For DeepSDF and DualSDF, we sample 500,000 points within the volume with annotated Signed Distance Function (SDF) as a pre-processing step. We test on 300 shapes that are distinct from their respective training sets. We observe that DeepSDF shows a promising reconstruction accuracy while DualSDF [3] is an order of magnitude worse. A possible deduction is that auto-decoder [1] is more suited than its variational counter-part [39]. At the same time, IM-Net and PQ-Net show slighty inferior reconstruction results. Comparison of different methods are summarised in Table 2.1 where the reconstruction accuracy is measured by two-way Chamfer Distance (CD), scaled by a factor of $10^{-}4$, where lower scores amounts to better reconstruction quality. We visualise one qualitative example in Figure 2.2 while we show more visualisations in the Appendix Figure A.1.

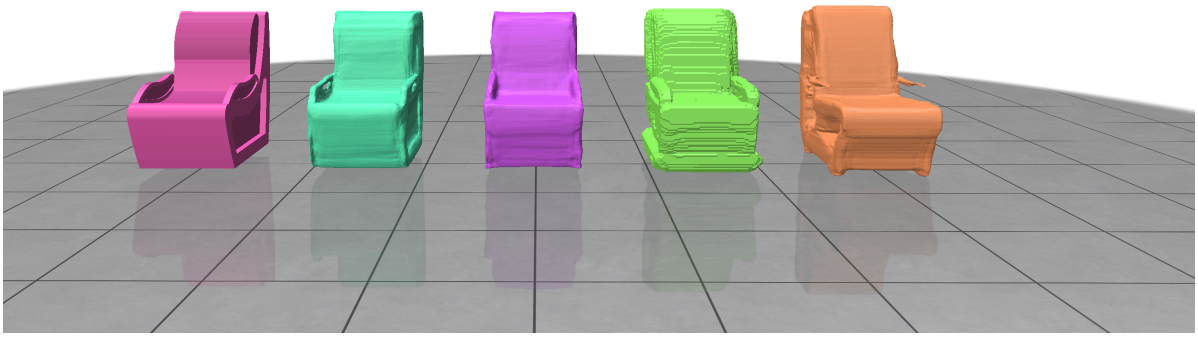


Figure 2.2.: Comparison between various reconstruction methods outlined above. Order of images are (from left to right) : Original, DeepSDF, IM-Net, PQ-Net and DualSDF.

Method	CD ↓
DeepSDF	3.2
DualSDF	30.4
IM-Net	4.9
PQ-Net	6.3

Table 2.1.: Comparison between different Deep Implicit Neural network for reconstructing Chairs from ShapeNetV2 dataset. ↓ denotes lower values are better preferred.

2.4. High-Frequency Reconstruction

While DeepSDF performed better than other reconstruction methods, it fails to reconstruct high-frequency mesh level textures and non-rigidly deforming surfaces as shown in Figure 2.4. Furthermore, DeepSDF considers all input points equally likely for predicting its SDF value. However, this induces a bias in point sampling density, thereby exacerbating reconstruction in thin-volume regions as visualised in the last row of Figure A.1. To address this, we borrow inspiration from the signal processing literature for better representation of high-frequency components and curriculum learning strategy for improving the geometric detail of reconstruction.

2.4.1. Fourier Feature Mapping

While Neural Networks are highly expressive functions, they have been empirically shown to be exponentially slow in learning higher-frequency components of the input signal [40]. This effect is more prominent in our case of coordinate based MLPs, where the eigenvalues corresponding to the kernel expressed by MLPs rapidly fall off [41, 42]. In order to overcome this spectral bias, drawing inspiration from Tanick *et al.* [42], we over-parametrise the input to our Coordinate MLP by mapping the input coordinates to DeepSDF $\in R^3$ to a higher dimensional frequency domain [43] as

$$\gamma(\mathbf{v}) = \left[a_1 \cos\left(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{p}\right), a_1 \sin\left(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{p}\right), \ldots, a_m \cos\left(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{p}\right), a_m \sin\left(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{p}\right)\right]^{\mathrm{T}} \quad (2.4)$$

Where, a_i and b_i are the amplitude and the frequency component of the signal respectively while $p \in R^3$ denotes the input coordinate. Both frequency and amplitude are fixed during the training phase and this mapping is performed as pre-processing step. In summary, we map the input to DeepSDF which initially was $\in R^{3+d}$ to $\in R^{m+d}$ where m $\gg$ 3.

2.4.2. Hyper-Network

The objective function of DeepSDF is a straightforward L1 loss over signed distance prediction. While this is simple and effective, additional supervision in-terms of Normal reconstruction can potentially result in higher quality reconstruction. Fortunately, surface normals can be *continuously* estimated along the surface as spatial gradient of SDF which in our case is a simple back propagation through DeepSDF. These higher order derivatives provide denser supervision and with a sinusoidal activation function [44], they have been demonstrated to better learn high-frequency features.

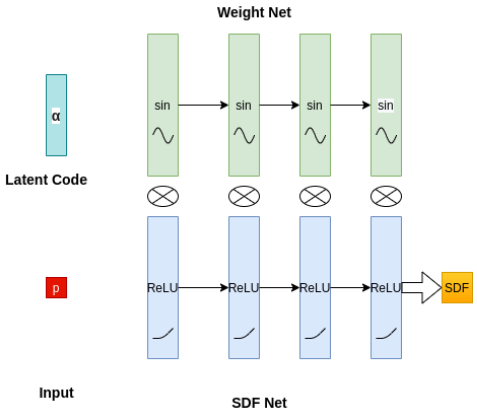


Figure 2.3.: Depiction of our HyperNetwork.

However, unlike the previous case of Fourier feature mapping, these constraints cannot be incorporated into DeepSDF. This is because DeepSDF entangles latent vectors to the coordinates as an input to the model and the input to such a network lies in R^{d+3} where d is the dimensionality of the latent vector. Thus, the dimensionality of point-wise normals would be in R^{d+3} preventing us from supervising the normal reconstruction.

To overcome this limitation, we draw inspiration from meta-learning [45, 46]. We disentangle the point coordinates and shape latent vector and our SDF prediction, resulting in two different networks for SDF prediction. The first network takes as input the coordinates of a point and predicts its SDF. The second network takes the shape latent code as input and predicts part of the weights of first network. Our architecture is depicted in Figure 2.3. At the training time, we optimise the following loss function,

$$\begin{aligned} \mathcal{L}_{\text{sdf}} &= \lambda_1 \int_{\Omega} |\nabla_{\mathbf{k}} f_\theta(\mathbf{k})| - 1| d\mathbf{k} + \lambda_2 \int_{\Omega_0} \|f_\theta(\mathbf{k})\| + \lambda_2 \int_{\Omega \setminus \Omega_0} \psi(f_\theta(\mathbf{k})) d\mathbf{k} + ||\alpha_i||_2 \\ \psi(\mathbf{x}) &= \exp(-\kappa \cdot |f_\theta(\mathbf{x})|) \end{aligned} \tag{2.5}$$

where, α_i is the latent vector of i^{th} shape, and $\kappa >> 1$. The first term corresponds to Eikonal boundary constraint, second term corresponds to supervising the point-wise normals, the third term penalises for wrongly classifying points on the surface, the fourth term penalises harshly for wrongly classifying points off the surface and the last term is used to promote compactness in latent space. At test time, the parameters of the network are fixed and we minimise the first four terms of the above loss function.

2.4.3. Curriculum DeepSDF

DeepSDF considers all input points equally likely in predicting its SDF. This can potentially lead to poorer reconstruction in areas of sparse point samples, thinner volumes, etc. Curriculum DeepSDF [47] proposes a twofold solution to overcome these shortcomings. First, to improve surface reconstruction accuracy, points are incrementally penalised for wrong SDF prediction. This implemented by using a threshold within which the DeepSDF is not penalised for a wrong prediction while gradually reducing this threshold to zero. Second, to address sampling discrepancy, a *weighted* loss function is used to penalise "hard", "semi-hard" and "easy" examples [48]. We adapt these supplemental losses into our reconstruction pipeline.

2.4.4. Experiments and Observation

We compare the aforementioned methods for high frequency reconstruction on two datasets containing high levels of naturally occurring textures. First, we consider bottles of different designs, which are proprietary dataset of Danone. High-frequency features appears in the form of ridges on the surface of the bottle. We use 61 meshes for training and 74 meshes for the test set. Second, we use the open-source dataset, MPI-FAUST [49], consisting of humans in different poses registered to a template mesh. This dataset consists of 100 meshes, with 10 distinct humans in distinct 10 poses. The first 80 meshes consisting of 8 subjects are chosen to be the training set and the latter 20, consisting of 2 distinct subjects are reserved for the test set.

We train all the models for a total of 2000 epochs on both the dataset, using ADAM optimiser [50] with a starting learning rate of 1e-3. Our model consists of 6 layered MLP with Weight Normalisation applied at each layer for DeepSDF based experiments. On the other hand, we empirically found out 4 layered MLP with weight normalisation to be optimal for Hyper-Network. For training our Hyper-Network 2.5, we use $\lambda_1 = 5e2, \lambda_2 = 1e2, \lambda_3 = 1e1, \lambda_4 = 3e3, \lambda_5 = 1e3$ and $\kappa = 3e3$. For experiments with Fourier feature mapping, we use amplitude a=1, frequency $b = 4096$ and map the coordinates to m=510 dimension. We use 500,000 sample points per shape for all experiments. For DeepSDF based experiments, 80% of points are sampled near to the surface and the points are annotated with its SDF

value [1]. For experiments with Hyper-Network we use 100,000 surface points along with surface normals and 400,000 points that are sampled within the volume close to the surface of the object. In this case however, the *signed* distance values of a point is not necessary and a binary representation denoting whether a point lies on or off the surface of the object is suffice. Reconstruction accuracy in terms of two-way Chamfer Distance scaled by 10^4 of aforementioned models across two datasets is summarised in the Table 2.2. *+FFM* denotes mapping the input coordinate $p \in \mathbb{R}^3$ to higher dimensional frequency domain as mentioned in Equation 2.4. CSDF denotes Curriculum Learning strategy applied to DeepSDF. We observe that Curriculum DeepSDF is the best performing method on the FAUST dataset while DeepSDF+FFM is the best performing method on the Bottles dataset. While Hyper-Network seems to better reconstruct the ridges of the bottles as depicted in Figure 2.4, it however produces artefacts such as holes on the surface as visible in the top portion of the reconstruction. We attempted to use more points on the surface and increase the capacity of the network, yet these artefacts persisted and its source is presently unclear. For the FAUST dataset, as shown in Figure 2.5 DeepSDF reconstructions are overly smooth while that of Curriculum DeepSDF reconstructs sufficient level of details accurately. We also observe that using Fourier Feature Mapping and Hyper-Network introduces artefact in reconstruction and fails to reconstruct mesh-level textures accurately in the FAUST dataset. We provide more visual examples in the Appendix section, in Figure A.2 for Bottles dataset and Figure A.3 for FAUST dataset.

Method	FAUST	Bottles
DeepSDF	6.8	4.2
DeepSDF + FFM	10.8	2.6
CSDF	5.8	2.9
CSDF + FFM	13.3	2.8
Hyper-Network	15.2	4.4

Table 2.2.: Comparison of Reconstruction accuracy of different methods on FAUST and Bottles dataset. Performance is measured by two-way Chamfer Distance.

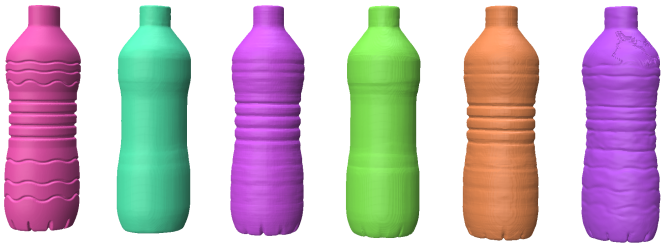


Figure 2.4.: Comparison between various high-frequency reconstruction methods outlined on the Bottles dataset. From left to Right : Original, DeepSDF, DeepSDF+FFM, CSDF, CSDF + FFM and Hyper-Network

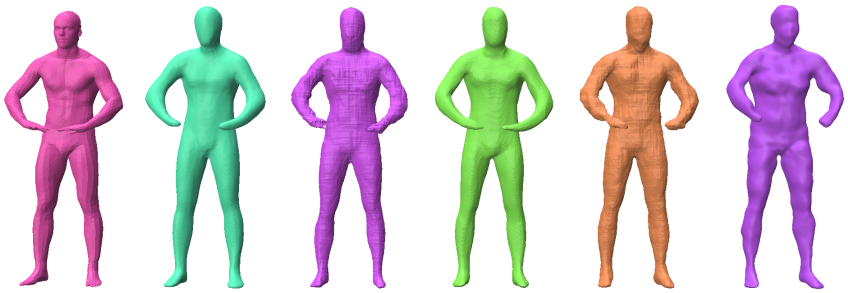


Figure 2.5.: Comparison between various high-frequency reconstruction methods on the FAUST dataset. Left to Right : Original mesh, DeepSDF, DeepSDF+FFM, CSDF, CSDF + FFM and Hyper-Network reconstructions

2.5. Summary and Remarks

In this chapter we have performed an extensive benchmarking of different Deep Implicit Neural Networks for high-fidelity 3D shape reconstruction. First, we compared auto-encoder with encoder-free implicit neural network and observed that encoder-free approach was better suited for our purpose due to better reconstruction quality. Then, we identified some well-known shortcomings with DeepSDF and explored two novel solutions for better reconstructing high-frequency components, namely by over-parameterising the input to the Implicit Neural Network by Fourier Feature Mapping. We benchmark different approaches for high-fidelity reconstruction on two different datasets, namely, the FAUST dataset consisting of articulated humans undergoing non-rigid deformation and Bottles dataset from Danone. We observed that Curriculum DeepSDF [47] was the best performing method on the FAUST dataset while DeepSDF with Fourier Feature Mapping was the best performing method on the Bottles dataset. We leverage these observations to use appropriate models in forthcoming chapters of this report.

3. Shape Optimisation

In this chapter, we use the implicit 3D reconstruction methods discussed in the previous chapter to build our novel shape optimisation pipeline. Unlike existing optimisation techniques that either operate on discrete surfaces with fixed connectivity or require large amount of training data, we show that our optimisation pipeline is robust in the presence of limited training data and be easily scaled to model multiple constraints.

3.1. Introduction

Properties of a 3D object can be either extrinsic, where the changes can be perceptually reasoned or intrinsic, where changes cannot be perceptually reasoned for. While modifying the former is straightforward - example shear and stretch, modifying the latter is non-trivial. Change in intrinsic properties such as static-friction, malleability, ductility etc. cannot be reasoned from perception alone but require additional feedback. In the design industry, such tasks often require the expertise of artists which in general are expensive and plodding. We observe in coherence with the recent literature of 3D Shape Analysis that such problems can be efficiently addressed using data-driven techniques. However, existing methods for optimisation are either task specific, susceptible to topological noises and require large amount of training data. In contrast, in this chapter, we develop a novel pipeline for optimising both intrinsic and extrinsic properties that are surprisingly simple yet highly effective in comparison to more sophisticated methods. We will show that with a well-learnt shape latent space, a shallow MLP is suffice for predicting physical properties from the shape latent vector with accuracy which is comparable with methods that learn directly from the geometry. We will also discussed a constrained optimisation setting, that is effective and robust in the settings where we have access to training data that is an order of magnitude smaller than the requirement of existing optimisation techniques. In summary, the overall goal of this chapter is to find a latent representation of a shape that satisfies one or more given property(ies). Recapitulating the relationship between latent vector and mesh from the previous chapter, first, the latent vectors are decoded into point-wise SDF within a volume. Then, the zero-level-iso-surface is extracted using Marching Cubes [33].

3.2. Preliminaries

In this section, we will briefly overview common technical terminologies used throughout this chapter.

3.2.1. LP-Net

Learning to predict the the intrinsic and extrinsic physical properties from the latent vector is an important step prior to optimisation. Our approach for prediction also has to be differentiable in order to be able to adapt to the optimisation pipeline. To this end, we use a 4-layered MLP which takes the shape latent vector, produced by DeepSDF and predicts different properties. This LP-Net is generic in its use and can be trained to predict one or more physical properties.

3.2.2. Topload

The physical property which we are the most interested in optimising is the Topload and is pertinent to bottles. It is defined as the vertical compression strength of a bottle and measured as an indicator of the bottle pallet behaviour. In a more intuitive sense, Topload can be visualised as a stability measure of bottles, typically for the ones made out of plastic. Such stability are measured through simulations performed on CATIA by applying a continuously increasing external forces at two vertical ends of the bottle. The maximum force at which bottle starts to corrugate is defined as its Topload. As it is a Force, Topload is measured in Newtons (N) and expressed in decaNewtons (dN) throughout this report for convenience sake. The simulation decouples geometry from physicality, wherein, different simulations performed on the same geometry with differing mass of the bottle yields a distinct Topload value.

3.3. Related Work

In this section, we recapitulate current literature in Shape Optimisation and methods for learning from 3D data. While the former is our direct goal, the latter consists of different state-of-the-art methods for Point Cloud classification and segmentation which we use to benchmark against our proposed LP-Net.

[0]https://www.3ds.com/products-services/catia/

3.3.1. 3D Shape Optimisation

Given a 3D model, the end goal of shape optimisation is to modify the *design* to optimise a particular physical property with minimalistic change in its geometry [51]. While an arbitrary 3D object has innumerous physical properties, the most explored ones in the context of shape optimisation are stability [52, 5, 53, 54], durability [55, 56, 57] and fluid-dynamic constraints [58, 59, 60]. With design engineering as the primary goal, first known data-driven approach for shape optimisation was explored by Funkhouser *et al.* [61], where the user provides design using sketches and the system retrieves similar geometry from a database. Since then, Deep Learning techniques have been increasingly used to emulate the environment [62, 63] and modelling physically plausible geometry [4, 5]. Recent and the most similar to our approach is that of Mezghanni *et al.* [5] who employ an implicit 3D model [35] for modelling shapes with improved stability. They use a differentiable loss based on persistent-homology [64] over the generated 3D-shape. In contrast to this, our approach performs optimisation in latent space. In another relevant data-driven approach, Baque *et al.* [59] use Geodesic Convolution Neural Network [65] to emulate fluid-dynamics simulator to optimise parts of aircrafts to satisfy certain aerodynamic constraints. We remark that such approaches are strongly susceptible to topological noises, while ours, thanks to implicit shape modelling is free-from dependence on connectivity.

3.3.2. Learning from 3D data

PointNet and PointNet++ [66, 67] are two well known and first Deep Learning based methods on point clouds data. They use a simple yet effective max pooling and spatial transformer networks [68] for local permutation invariance. Following this, KPConv [69] attempts to reproduce the working principle of CNNs on Point Cloud. They apply Convolution Operation on Points, where the convolutional filters are located in continuous Euclidean space learnt from data. To ameliorate the fact that point clouds inherently lack topological information, Dynamic Graph CNN [70](DGCNN) "dynamically" constructs a graph at each layer in a differentiable manner while aggregating neighbouring information in the graph using a novel EdgeConv module. DiffusionNet [71] is yet another method for learning from PointCloud and Meshes, simple in built and comprehensive in terms of applications. It learns point-wise diffusion (propagation of heat on a surface) through learnable diffusion layer and builds anisotropy through gradient features by projecting the estimated point-wise normals to the tangent plane. Learning diffusion and anisotropy are built to be analogous to mimic convolution and pooling operation on 2D images. In the context of this report, we use the aforementioned methods to benchmark the efficacy of our proposed LP-Net.

3.3.3. Latent space shape exploration

At the heart of all data-driven techniques lies a canonical representation of data in lower dimension. Such low-dimensional subspaces have been well-studied in the context of shape space exploration even prior to the advent of Deep Learning [72, 73, 74, 75] with a particular focus on leveraging symmetry [76, 77]. Shapira *et al.* [72] extract low-dimensional shape embedding using a mixture of Gaussian models in the seminal work on data-driven latent space exploration. In the recent literature, such lower dimensional representation from a Deep Network, typically latent vectors, are being extensively studied for learning mesh deformation space [4, 78, 79], cage deformation space [80], vertex offsets [81] etc. However, to the best of our knowledge, currently Latent space shape exploration has not been studied in the context of 3D shape optimisation of physical properties. We believe our work is the first to study the applicability of learnt shape latent space for Shape Optimisation.

3.4. Approach

In this section, we delineate different approaches for optimising 3D shape at its latent space. We first demonstrate our straightforward gradient ascent based approaches and then discuss constrained latent space optimisation. While our first approach is simple and straightforward, such techniques often do not scale well with Deep Learning methods due to the highly non-convex nature of the optimisation space. Furthermore, our goal is to also build an approach that is robust in settings where paucity of training data is a key concern. To this end, we delineate new constrained optimisation techniques, where we force the optimisation to be in the linear subspace defined by the convex-hull of latent vectors. By carefully choosing points that define this convex-hull, we empirically observe, as shown in the subsequent section, that our approach converges to a minimum.

3.4.1. Generic Latent Space Optimisation (LSO)

Using the pre-trained LP-Net and DeepSDF, the goal is to sequentially modify the latent vector to generate new shapes which are realistic while faithfully reflecting the change in physical property of the object of interest. Let f_θ denote trained Deep-SDF with fixed parameters θ, g_ϕ denote the LP-Net with fixed parameters ϕ and the latent vector denoted by $\vec{\alpha}$. Then, the task of LSO is to find the optimal latent vector by minimising the following energy,

$$\mathcal{E} = \underset{\vec{\alpha}}{\operatorname{argmin}} \;\; \lambda_1 \Big| \sum_{i=1}^{N} f_\theta(x_i, \vec{\alpha}) - s_i \Big| + \lambda_2 |g_\phi(\vec{\alpha}) - \delta| \tag{3.1}$$

Where δ is the *scalar* physical property of interest. The parameter λ controls the impact of LP-Net in modifying $\vec{\psi}$. The first term of this energy enforces geometric similarity through

point-wise SDF while the second term optimises a physical property.

3.4.2. Dual Latent Space Optimisation (D-LSO)

Optimisation discussed for a single scalar property in the previous section can be extended for multiple properties in a straightforward manner. For simplicity, we delineate how this optimisaiton can be done for two properties simultaneously. Let g_{ϕ_1} be LP-Net with parameters ϕ_1 trained to predict the scalar property δ_1 and g_{ϕ_2} be LP-Net with parameters ϕ_2 trained to predict the scalar property δ_2. Then, the task of dual-LSO is to minimise the following energy,

$$\mathcal{E} = \underset{\vec{\alpha}}{\operatorname{argmin}} \;\; \lambda_1 | \sum_{i=1}^{N} f_\theta(x_i, \vec{\alpha}) - \sigma_i | + \lambda_2 | g_{\phi_1}(\vec{\alpha}) - \delta_1 | + \lambda_3 | g_{\phi_2}(\vec{\alpha}) - \delta_2 | \tag{3.2}$$

3.4.3. Adversarial Latent Space Optimisation (A-LSO)

Additional constraints can be imposed on the latent vector to lie in the realistic shape space. For this, we use the Discriminator of L-GAN [11] to classify the latent vector to be real or fake. First, we train the L-GAN to classify shape latent vectors produced by our reconstruction method as real and ones generated by the Generator to be fake. Then, at the time of optimisation, we use this mis-classification penalty from this pre-trained Discriminator to encourage latent vectors to be realistic. Considering h_ψ to be the Discriminator with learnt parameters ψ, the goal of our new adversarial LSO is then,

$$\mathcal{E} = \underset{\vec{\alpha}}{\operatorname{argmin}} \;\; \lambda_1 | \sum_{i=1}^{N} f_\theta(x_i, \vec{\alpha}) - \sigma_i | + \lambda_2 | g_{\phi_1}(\vec{\alpha}) - \delta_1 | + \lambda_3 | 1 - h_\psi(\vec{\alpha}) | \tag{3.3}$$

3.4.4. Local Latent Space Optimisation (L-LSO)

The optimisation which we have seen until now can quickly converge to a local minimum, where the latent vector would not correspond to a meaningful shape, for example, the extracted iso-surface might not lie in the volume range. This is bound to happen when there is insufficient training data and the latent vectors are far from each other in the embedding space. Leveraging on the previous observations [1] that shape latent space is well defined along the linear path between two latent vectors, we propose a new retrieval based local shape optimisation approach. Let us assume that we are given a shape $\mathcal{S}_0$ with associated latent vectors and physical property α_0 and δ_0 respectively. Our goal is to optimise the latent vector α_0 to $\tilde{\alpha}_0$ whose physical property is close to δ_1 while the optimised shape being visually similar to $\mathcal{S}_0$. Our algorithm can then be defined as follows,

Algorithm 1 Retrieval based local shape optimisation

Require: $\mathcal{S}_0, \alpha_0, \delta_0, \delta_1, K$
Ensure: $\{\alpha_0 \dots \alpha_N\}$
Compute L_0 corresponding to S_0
Set $j = 0, \mathcal{T} = \emptyset$
while $j \leq K$ **do**
    Find $L_i = \underset{\vec{\alpha}}{\text{argmin}} |g_\phi(\vec{\alpha}) - \delta_1| + ||L_j - L_0||_2 \quad \forall L_j \in \mathcal{L}$
    $L_k = t\dot{L}_i + (1-t)\dot{L}_0 \quad$ where, $t \in (0,1)$
    $\mathcal{T} = \mathcal{T} \cup L_k$
    $\mathcal{L} = \mathcal{L} \setminus L_i$
    $j+ = 1$
end while
return $\mathcal{T}$

3.4.5. Optimisation in a convex region

The constraint we imposed in the previous case is too strict in the sense that we can only generate shapes that lie along the path traced between query shape and the retrieved shape. This however limits the generative capability of our optimisation. To alleviate this shortcoming, we propose to find the optimial latent vector in the region defined by convex hull of K-points. For mathematical brevity, we restrict K=4 to our discussion, but theoretically K can take any value between K=1 and K=M, where M is the size of training shapes. To recap, K=1 was discussed in the previous section. Our objective for optimisation in convex region is then defined as

$$\begin{aligned} \mathcal{E} =& \underset{\vec{\Omega}}{\text{argmin}} \sum_{i=0}^{N} \lambda_1 \left| f_\theta\left(\Omega, x_i\right) - s_i \right| + \lambda_2 \left| g_\phi(\Omega) - \delta_1 \right| \\ \text{Where, } \Omega =& w_1\vec{\alpha 1} + w_2\vec{\alpha 2} + w_3\vec{\alpha 3} + w_4\vec{\alpha 4} \\ & \text{s.t } w_i \in [0,1] \text{ and } \sum_{i=1} w_i = 1 \end{aligned} \tag{3.4}$$

Unlike the case of optimisation in Equation 3.1, where the latent vectors are updated through gradient ascent, in this case, we only update the variables highlighted in green in the above equation to restrict the latent vector to lie in the convex-hull.

3.4.6. Optimisation in a convex region with additional constraints

While optimising in the convex region imposes constraint on the space where latent vector is optimised, it still comes with the same flexibility as the standard LSO and allows us

to add more constraints to the latent space in-terms of the scalar property we predict. In our experiments, we use volume constraint in addition to the Topload constraint while encouraging realism through the adversarial loss similar to Equation 3.3. Considering g_{ϕ_2} to be LP-Net with parameters ϕ_2 trained to predict the scalar property δ_2, then, the dual-constraint optimisation in convex region is,

$$\begin{aligned}\mathcal{E} =& \underset{\vec{\Omega}}{\operatorname{argmin}} \sum_{i=0}^{N} \lambda_1 \left| f_\theta\left(\Omega, x_i\right) - s_i \right| + \lambda_2 \left| g_{\phi_1}(\Omega) - \delta_1 \right| + \lambda_3 |g_{\phi_2}(\vec{\alpha}) - \delta_2| + \lambda_4 |1 - h_\psi(\vec{\alpha})| \\ \text{Where, } \Omega =& w_1 \vec{\alpha 1} + w_2 \vec{\alpha 2} + w_3 \vec{\alpha 3} + w_4 \vec{\alpha 4} \\ & \text{s.t } w_i \in [0,1] \text{ and } \sum_{i=1} w_i = 1\end{aligned} \tag{3.5}$$

Where h_ψ is the Discriminator with fixed parameters ψ.

3.5. Results

We first explore different scalar properties of solids and illustrate the robustness of LP-Net. To justify our previous claim that well-learnt latent vectors are capable of accurately predicting intrinsic scalar properties, we exhaustively benchmark LP-Net against other state-of-the-art methods that operate on Point Clouds and Meshes for comparing intrinsic properties. We then demonstrate qualitative and quantitative results for shape optimisation. Our experiments are mainly focused on bottles from Danone while we also demonstrate preliminary results on Chairs from the ShapeNetSM [82] benchmark.

3.5.1. LP-Net Performance

We start with a relatively easy task of predicting scalar property using LP-Net. While this is not our primary goal, it is important to have an optimal prediction as this step plays a crucial role in gradient based shape optimisation.

Extrinsic Property prediction

We show the efficacy of LP-Net for predicting the extrinsic scalar properties namely Width, Height and Length and Weight. We use a 3 layer MLP for LP-Net with weight normalisation and 256-d hidden size. The prediction accuracy (in terms of Mean Square Error) is summarised in Table 3.1. We observe an overall convincing performance while admitting a minor over-fitting in weight prediction.

Scalar to Predict	Accuracy (Training set)		Accuracy (Test set)	
	Bottles	Chairs	Bottles	Chairs
Width	93%	94%	94%	88%
Height	95%	96%	96%	91%
Length	94%	93%	91%	85%
Weight	94%	84%	95%	79%

Table 3.1.: Comparing performance of LP-Net on Chair class from ShapeNet-SM dataset and Bottles dataset.

Topload Prediction

Topload is an intrinsic property of bottle that we are interested in optimising. To this end, we need to have a good estimation of topload, ideally with our LP-Net. In this sub section, we will compare the Topload prediction of LP-Net with other state-of-the-art point cloud processing methods which are typically used for segmentation and classifying tasks. Ground truth Topload is obtained by performing simulation where a bottle is given an arbitrary mass that is distributed along the vertices of a mesh. For mesh and point cloud based methods, we will use this additional point-wise information. On the other hand, for LP-Net, which predicts Topload from the latent vector, we will concatenate the sum of all the point-wise mass to the latent vector.

We benchmark different methods for Topload prediction on the bottle dataset in Table 3.2. For PointNet, we use 5 fully connected layers whereas for PointNet++ we use three-level hierarchical network with three fully connected layers and kNN based neighborhood search with k=64 neighbours. For DynamicGraphCNN we use 3-layered MLP stacked with dynamic EdgeConv module as feature extractor with k=30 nearest neighbors in the feature space to dynamically construct the graph. Among multiple variants for DiffusionNet, we found 20,000 vertices per shape per batch with 192 filters and 8 diffusion blocks to have the best test set accuracy. For PointNet++, PointNet, DGCNN and DiffusionNet, the input to the network consists of mesh vertices stacked together with point-wise thickness values. DiffusionNet-Mesh denotes the same DiffusionNet model applied on the mesh instead of PointCloud. In this case, however, we do not use point-wise thickness values as the triangulation do not share same vertices as before. We posit this to be the reason for inferior performance than its Point Cloud counterpart. To use MeshCNN [27] on our meshes, we first apply quadratic edge collapse [83] to simplify the existing mesh to contain approximately 2k edges. Then we apply 4 MeshConv operations followed by MeshPool operations and finally apply Global Average Pooling followed by 2 MLPs to predict the Topload value. For our LP-Net, we empirically found out that 4-layered MLP produced the best results. We compare two different variants of LP-Net, one with weights of the bottles (in grams) stacked to its latent vector denoted as

LP-Net and without the weight information denoted by LP-Net w/o weight.

Method	Accuracy (training set)	Accuracy (Test set)
PointNet [66]	84%	83.5%
PointNet ++ [67]	85.5%	86%
Dynamic Graph CNN [70]	82%	85%
DiffusionNet [71] (Point Cloud)	90%	85%
DfN [71] (Mesh)	82%	79%
MeshCNN [27]	82%	77%
LP-Net w/o weight	88%	81%
LP-Net	87%	84%

Table 3.2.: Comparing various aforementioned point-based learning methods on bottle dataset for Topload prediction.

3.5.2. Extrinsic Property optimisation

In this sub-section we illustrate quantitative and qualitative results of optimising extrinsic scalar properties, namely length, width, height and weight on bottles and chairs dataset. We believe this serves as an empirical proof-of-concept for intrinsic property modification. A proof-of-concept is necessary as unlike the case of extrinsic property optimisation, we do not have a visual feedback or can measure the change in the property in the case of intrinsic optimisation.

We first compare between the generic LSO described in Equation 3.1 and Adversarial Latent Space Optimisation (A-LSO) described in Equation 3.3. We measure the optimisation based on two criteria - first, reconstruction accuracy as two-way Chamfer Distance (CD) multiplied by 10^3 and second, the measured difference in scalar property (Ratio). Results for change in height, width and length as a result of LSO are summarised in Table 3.3 and we demonstrate a qualitative example in Figure 3.1.

Method	CD↓			Ratio of Change		
	Height	Width	Length	Height	Width	Length
Direct Reconstruction	0.94			1.00		
Increase	3.5	2.8	3.2	1.18	1.10	1.09
Increase + GAN	3.4	2.6	2.9	1.18	1.10	1.08
Decrease	4.3	4.1	4.4	0.89	0.90	0.85
Decrease + GAN	4.1	4.1	4.2	0.89	0.90	0.85

Table 3.3.: Extrinsic scalar property modification. +GAN denotes the use of adversarial loss as described in Equation 3.3

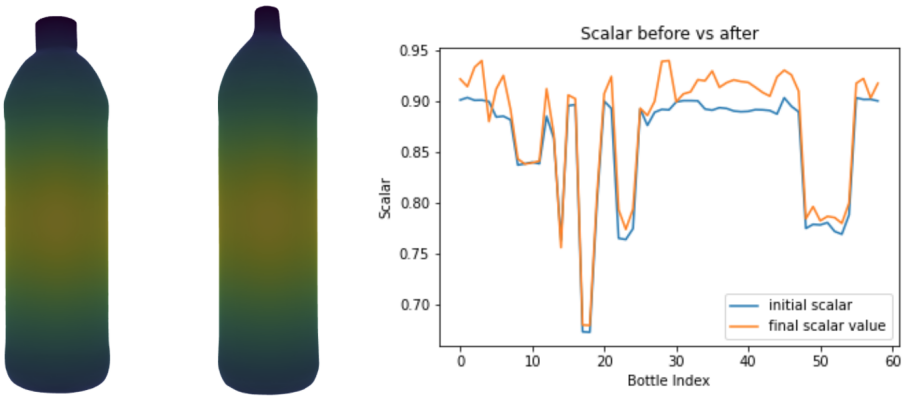


Figure 3.1.: From Left to Right : Original bottle, change in height by LSO and magnitude of change in the height across the test set.

3.5.3. Topload Optimisation

Optimising the Topload of bottles is our main goal. Given a bottle and optionally its current Topload measure and the target Topload, the goal of this optimisation is to modify the geometry of the bottle to satisfy the target Topload while remaining visually similar to the given bottle. For all our experiments, the end goal is to optimise Topload to 170dN. However, unlike the experiments discussed before, in this case there is no unique well-defined metric. Therefore for quantitative comparisons, while we measure the similarity to the starting mesh, it is important to keep in mind that a lower Chamfer's Distance is a necessity but not sufficient condition. Additionally, we also report the Minimum Matching Distance (MMD) [11] to gauge its fidelity. We compare on the 74 test set meshes used in the previous section for reconstruction. The quantitative performance results of optimisation is summarised in the Table 3.4. ↑ denotes that higher value of the metric is preferred while ↓ denotes that lower

value of the metric is preferred. Our rule of thumb to gauge different approach is based on lower MMD, higher Mean Topload and lower change in volume.

We denote the standard Latent Space Optimisation corresponding to Equation 3.1 as LSO and with adversarial loss described in Equation 3.3 as A-LSO. Local Latent Space Optimisation as depicted in Algorithm 1 is denoted as L-LSO. C-LSO denotes the dual-constrained Latent Space Optimisation as described in Equation 3.5 typically performed with k=3 neighbours. We re-iterate that the choice of neighbour is a crucial step in our optimisation. We choose at random, one neighbour from the training set whose Topload value is larger than 150. Empirically, we have observed this to be the best practise while choosing neighbours whose Topload values are smaller than the query shape often fails to converge. "C-LSO NN" is the case when two neighbours are chosen based on *minimum* area formed by the triangle in the latent space while in the case of "C-LSO FN", two neighbours are chosen such that area of the triangle in the latent space is maximum. "C-LSO W/O Volume" refers to the optimisation described in Equation 3.4. Finally, we also compare to the case when we optimise in the Convex Hull defined by four points in the latent space as "C-LSO k=4".

For all the experiments, we perform optimisation for 4000 steps or until convergence using ADAM optimiser [50] with a learning rate of 0.001. We empirically observed that using $\lambda_1 = 1, \lambda_2 = 0$ for first 800 iterations and then fix $\lambda_1 = 0.1, \lambda_2 = 0.05$ gives best performance in-terms of reconstruction. This is because reconstruction of the shape in the first 800 iterations is important prior to optimisation. For experiments with A-LSO, we use $\lambda_3 = 0.01$ the coefficient for adversarial penalty and use W-GAN GP [15]. For optimisation in convex region, we empirically found out that initialising the latent vector *close* to the query latent vector proved effective in convergence. To enforce the convexity constraint in a differentiable manner, we apply softmax over the set of coefficients w_is. Throughout the study, our primary constraint is the Topload and secondary constraint whenever used refers to the Volume of the bottle.

For the vanilla Latent Space Optimisation denoted as LSO constantly converges to a local minimum, meaning, the latent vector does not render an iso-surface upon extraction using Marching Cubes for the majority of shapes in the test set. Thus, we do not report any metrics for the same. We posit insufficient training data to be the reason for this convergence to local minimum. A-LSO seems to be the best performing method in terms of fidelity while C-LSO NN compromises the reconstruction fidelity to further optimise Topload. When the constraint on the volume is not applied, the Topload is better optimised, however, there is a significant change in the volume than desired. We show one qualitative example of our optimisation in Figure 3.2 while we visualise more examples in the Appendix section in Figure A.4.

Method	CD↓	MMD↓	Mean Topload (dN) ↑	Change in Voume (cL)↓
LSO	-	-	-	-
A-LSO	0.29	2.9	140.1	45.0
L-LSO	20.9	11.1	131.7	97.2
C-LSO NN	19.2	9.0	157	45.8
C-LSO FN	72.9	31	165	50.4
C-LSO W/O volume	14.0	8.7	163	94.5
C-LSO k=4	12.4	0.7	120.2	61.4

Table 3.4.: Qualitative comparison of different variants of LSO.

Figure 3.2.: Comparison of different approach for Topload optimisation. From left to right : A-LSO, L-LSO, C-LSO NN, C-LSO with k=4 neighbours.

3.6. Summary and Remarks

In this chapter we introduced our novel Latent Space Shape Optimisation (LSO) along with many variants to optimise properties of shape in its latent space. We also demonstrated that using a light-weight 4-layered MLP on the shape-learned latent vector can predict extrinsic (visually perceptible) physical properties with very high accuracy. Complimenting this, we also demonstrated that predicting intrinsic shape property, Topload of a bottle from this latent vector performs comparable to approaches that use surface-level information. We then extensively benchmark the different variants of Latent Space Optimisation (LSO) and empirically found out that without additional constraints imposed on the latent space, our optimisation does not converge. The reason for this phenomenon is that when there is insufficient training data, the latent vectors are well separated in the latent space and gradient ascent alone is incapable of attaining the global minima due to highly non-convex nature of the optimisation space. To ameliorate this shortcoming, we proposed constrained optimisation in the convex hull defined by "k" other training set shapes. Empirically we demonstrated that using k=2 neighbours is the best performing method in terms of final Topload optimisation value and at the same time better respects the volume preservation constraint.

While this approach is interesting there are a few known limitations. First, we do not have an exact metric to gauge the optimisation and rather use an amalgam of several measurements to gauge the optimisation. Second, limitation is with respect to fixing the neighbours prior to the optimisation process and requiring one of the neighbours to have a larger Topload value than the query shape. Using learning based techniques [4] to overcome this limitation is an interesting future work to ponder upon. Finally, an important point of research is to better understand the failure of LSO. At present, extracting a shape from its latent representation involves estimating point-wise Signed Distance Field (SDF) within a voxel of arbitrary volume and applying Marching Cubes [33] to extract the iso-surface. This process is non-differentiable and feedback from visual inspection is limited. Therefore, to have a better theoretical understanding of such failures and devising appropriate constraints in our optimisation is a key area of research for future work.

4. Non Rigid Shape Correspondence

In this chapter, we propose a new non-rigid shape correspondence approach based on latent neural networks which we used for reconstruction purposes in the previous chapters. We demonstrate that latent vectors from implicit neural networks can be successfully adapted to template based deformation approaches for shape correspondence and are robust to strong artefacts, including significant noise and missing data. As the direction of research we discuss in this chapter in itself is novel, we conclude this section by precise limitations of our current approach and potential future work.

4.1. Introduction

Given two non-rigidly deforming objects, a source and a target represented as a Tri-Mesh, our goal is to establish a map between each point on the source and the target. Non-rigid shape correspondence is a long standing central problem in Computer Vision and the expansive literature in this topic focuses on developing local descriptors [84, 85], representing shapes in a reduced basis [86], estimating a map in this reduced basis [6], and methods for map refinement through iteration [8, 7, 87]. All these methods are purely intrinsic, meaning, they do not leverage information on the location of point-wise coordinates of shape in the 3D space. On the other hand, a purely extrinsic approach in this domain tackles this problem by learning to deform a fixed template to establish correspondence between a source and a target. At present, such extrinsic methods [88, 89] are not sufficiently explored as they require multiple optimisation steps to learn ordering, rotation invariance and establish correspondence. Our key observation here is that a well-learnt latent representation of 3D objects can serve to be better suited for template deformation.

To this end, we propose to use shape latent vectors from Deep Implicit Neural Networks for learning to deform templates. Our proposed method is purely extrinsic and does not depend on connectivity, mesh density and can be adapted to match shapes with noise. Furthermore, we also demonstrate encouraging results in partial shape correspondence settings. This work is substantially a work in progress and we focus only on encouraging preliminary results and not robust findings. Our results show promising signs of the first step in implicit shape matching and open possibilities to explore methods which can potentially pave the way for combining intrinsic and extrinsic methods for non-rigid shape correspondence.

4.2. Related Work

To the scope of our discussion, the literature of non-rigid shape matching can be broadly divided into classical methods which relies on local information such as mesh connectivity and reconstruction based methods which learns correspondence from template.

4.2.1. Classical Methods

Classical methods for non-rigid shape matching or more broadly shape analysis are intrinsic i.e, they leverage local connectivity of points [90] rather than considering their location in 3D space. Significant work in the past have designed various point-wise signatures which typically are invariant under rigid transformation [91, 92], non-rigid transformation [93] by preserving isometry [94], near-isometry [95, 6] and conformity [96, 97] to name a few. Recent data-driven spectral methods [98, 99, 100] in non-rigid shape matching follows the Functional Map pipeline [6], which is a linear transformation (represented by matrix) between functions defined on surface (ex. heat diffusion [85]) between source and target. The most interesting aspect is that these functions can be compactly represented in their respective basis, typically first k-eigen functions of the cotangent Laplacian matrix [101]. Recent advancement relevant to our current discussion includes techniques for map refinement [7, 8], leveraging learning based techniques [99, 98, 102] and inclusion of extrinsic information into the Functional Map pipeline through point-wise coordinates and normals [9, 10]. While Eisenberger*et al.* [10] show that extrinsic information can be built into Functional Map pipeline to disambiguate issues pertaining to symmetry, they use pre-computed point-wise signatures [103] that are highly sensitive to connectivity and meshing. We conclude this section by remarking that attempts insofar attempt to include discrete extrinsic information while our work can potentially pave a way to define such intrinsic quantities such as vector field inside a continuous volume.

4.2.2. Correspondence through reconstruction

Most reconstruction based approaches deform a shape template to match a given source and target [104, 88, 105, 106]. These templates are triangular meshes with fixed number of vertices, which are curated to a particular object (humans, animals, etc) and are carefully parametrised handle pose variations. Templates are deformed with two primary objectives 1) preserve order and 2) minimise reconstruction loss. Most notable and similar to our proposed approach is 3D-Coded [88]. They use a PointNet [67] based encoder to learn a latent representation of each object and deform each vertex of the template to match the given shape using a supervised loss function that necessitates the template to be in 1-1 correspondence with training shapes. At test time, source and target meshes are deformed independently and dense correspondence is established through nearest-neighbour search on the template. More

recently, two concurrent approaches propose to match shapes in implicit field [107, 106]. They respectively learn deformation in the implicit domain [1], by simultaneously learning a SDF for template as well as objects through point-wise transformation. The deformation in the implicit field however does not respect a 1-1 correspondence, have a strong normal consistency assumptions thereby making them unviable for the task of non-rigid shape correspondence.

4.3. Approach

Given a pair of shapes $\mathcal{M}$ and $\mathcal{N}$, typically represented as triangular meshes, our goal is to establish a point-to-point (P2P) map Π between each points $p_{\mathcal{M}} \in \mathcal{M}$ and $p_{\mathcal{N}} \in \mathcal{N}$. Let $\vec{\alpha}_{\mathcal{M}}$ and $\vec{\alpha}_{\mathcal{N}}$ be their respective latent representation obtained from our deep implicit reconstruction network f_θ. Let $\mathcal{T}$ be the template mesh with N vertices, which we deform using Displacement Network d_ψ with parameter ψ to establish correspondence.

Our first step involves learning to deform a template to match reconstruct the source and the target. The displacement network consists of 4-layered MLP which takes the shape latent vector $\vec{\alpha}$ concatenated with points from the template mesh as $\vec{\alpha} \oplus p_{\mathcal{T}}$ and outputs points in 3D space $\tilde{p}_{\mathcal{T}} \in \mathbb{R}^3$. The points $\tilde{p}_{\mathcal{T}}$ represent ordered vertices of the mesh which we aim to reconstruct. Training objective of the displacement network can be written as,

$$\mathcal{E} = \underset{\vec{\psi}}{\operatorname{argmin}} \; || \sum_{i=1}^{N} d_\psi(\vec{\alpha} \oplus p_i) - s_i||_2 \tag{4.1}$$

Where s_i is a point on the mesh that we aim to reconstruct, which are registered with the template mesh. Once the displacement network d_ψ is trained, we fix the parameters ψ to be fixed and optimise the template reconstruction by minimising the two-way Chamfer's Distance, to obtain the optimal latent vector as,

$$\begin{aligned}
\mathcal{L}_{\mathcal{M}} &= \underset{\vec{\alpha}}{\operatorname{argmin}} \; \sum_{\mathbf{p}\in\mathcal{M}} \min_{\mathbf{q}\in\mathcal{T}} \left|d_\psi(\vec{\alpha_{\mathcal{M}}} \oplus p_i) - \mathbf{q}\right|^2 + \sum_{\mathbf{q}\in\mathcal{T}} \min_{\mathbf{p}\in\mathcal{M}} \left|d_\psi(\vec{\alpha_{\mathcal{M}}} \oplus p_i) - \mathbf{q}\right|^2 \\
\mathcal{L}_{\mathcal{N}} &= \underset{\vec{\alpha}}{\operatorname{argmin}} \; \sum_{\mathbf{p}\in\mathcal{N}} \min_{\mathbf{q}\in\mathcal{T}} \left|d_\psi(\vec{\alpha}_{\mathcal{N}} \oplus p_i) - \mathbf{q}\right|^2 + \sum_{\mathbf{q}\in\mathcal{T}} \min_{\mathbf{p}\in\mathcal{N}} \left|d_\psi(\vec{\alpha_{\mathcal{N}}} \oplus p_i) - \mathbf{q}\right|^2
\end{aligned} \tag{4.2}$$

Different to Equation 4.1, here we optimise for the latent vector $\vec{\alpha}$ and not the parameters of the network. Then a point $p_{\mathcal{M}} \in \mathcal{M}$ and $p_{\mathcal{N}} \in \mathcal{N}$ are in correspondence if

$$\begin{aligned}
p_{\mathcal{M}} &= \underset{\mathbf{p}'\in\mathcal{T}}{\operatorname{argmin}} \; |d_\psi(\vec{\alpha}_{\mathcal{M}} \oplus p') - p_{\mathcal{M}}|^2 \\
p_{\mathcal{N}} &= \underset{\mathbf{p}'\in\mathcal{T}}{\operatorname{argmin}} \; |d_\psi(\vec{\alpha}_{\mathcal{N}} \oplus p') - p_{\mathcal{N}}|^2
\end{aligned} \tag{4.3}$$

In summary we use shape latent representation $\vec{\alpha}$ from an auto-decoder, which is primarily reconstruction network in contrast to using a PointNet encoder as in the case of 3D-Coded [88].

4.4. Experiments

In this section, we will illustrate the quantitative and qualitative performance of our method for the task of non-rigid shape matching across four different settings, namely, generic, a re-meshed dataset and on shapes with noise in the form of cuts and holes. In the case of data-driven methods, we train on the first 80 meshes of the MPI-FAUST dataset [49]. The performance is measured in terms of geodesic distortion which is the geodesic error computed on the surface of the target mesh between the ground truth corresponding point and predicted correspondence [108]. Methods with smaller geodesic error is gauged better than its counterpart. More formally, the metric is defined as follows,

Definition 4.4.1 (Geodesic Distortion) *Let $p_m \in \mathcal{M}$ and $p_n \in \mathcal{N}$ be points on shapes $\mathcal{M}$ and $\mathcal{N}$ respectively. Let $\tilde{\Pi}$ and Π be the predicted and ground-truth point-to-point map between $\mathcal{M}$ and $\mathcal{N}$ respectively, then the geodesic distortion [108] is computed as*

$$\mathbf{d} = d^{\mathcal{N}}(\tilde{\Pi}(x), \Pi(y)) \forall x, y \in M \tag{4.4}$$

where $d^{\mathcal{N}}(\cdot, \cdot)$ is the geodesic distance between points on shape $\mathcal{N}$.

4.4.1. Experimental Setup

We compare our method with two data-driven methods and two axiomatic methods. For axiomatic methods, we use Functional Map [6] with 20 point-wise WaveKernel [84] descriptors and 100 Eigenfunctions on source and target respectively. In addition we also use the orientation preservation operator [8] and multiplicative operators [109] in solving for the Functional Map. We refer to this method as BCICP in our comparisons. We refine the initial map produced by BCICP with 15 ZoomOut iterations, which we denote as "zoomout" in our experiments. We also compare with two learning based techniques, namely DeepShells [10] and 3D-Coded [88]. We pre-compute 352 dimensional SHOT descriptors [103] on each shape and use 200 Eigenfunctions for the truncated spectral filters and train on 80^2 training meshes from FAUST [49] dataset for 20 epochs. We train our method and 3D-Coded on 80 training meshes from [49] dataset for 1000 epochs to learn point-wise ordering - Equation 4.1 and 3000 iterations for latent code optimisation - Equation 4.2 respectively.

4.4.2. Generalisation to connectivity

In this sub-section we compare different methods on the original MPI-FAUST dataset [49] and the FAUST Re-meshed dataset [8]. For data-driven methods, we train only on the first 80 meshes of the MPI-FAUST dataset and evaluate on the last 20 meshes of both the datasets. Our quantitative results are summarised in Figure 4.2 and in Table 4.1. Method with highest percentage of correspondence at the cost of minimal geodesic error is gauged to be better performing. Our method performs the best on the FAUST Re-meshed dataset while it performs comparable to other methods on FAUST Original. A key observation here is that the geodesic distortion of other methods are nearly twice as worse as our method on the FAUST Re-meshed dataset implying they are sensitive towards connectivity. In contrast, ours and 3D-Coded does not show signs of deterioration with re-meshing. We visualise more examples of correspondence in the Appendix Figure A.5. We also show a qualitative comparison between correspondence predicted by our method and 3D coded along with the ground truth correspondence in Figure 4.1. Please note that ground-truth correspondence is not a bijective map. Hence, the grey regions denote the vertices on the target mesh that does not have any correspondence in the source mesh.

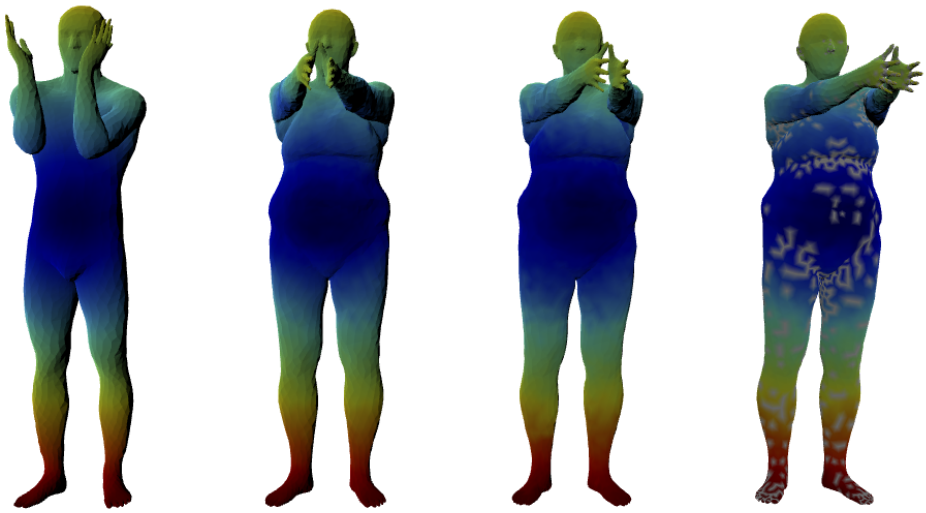
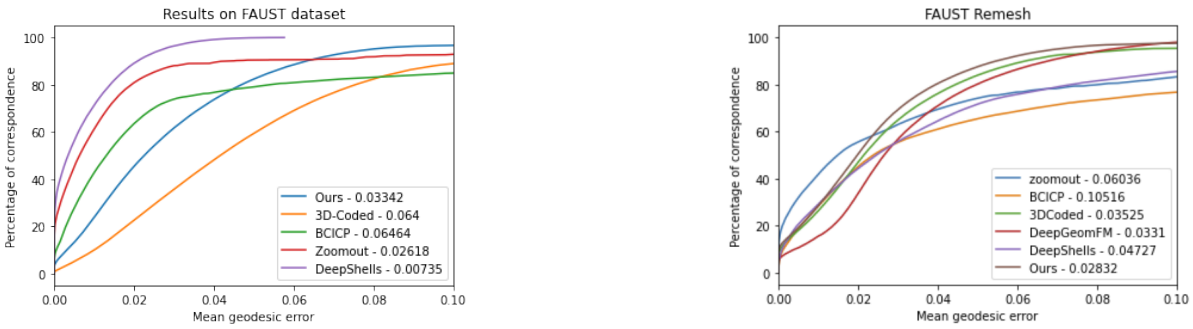


Figure 4.1.: Left to Right : 1) Source mesh, Target mesh with correspondence color-coded by 2) 3D-Coded, 3) Ours and 4) ground truth. Grey regions denote vertices with no Ground Truth correspondence as the ground truth map is *not* bijective.

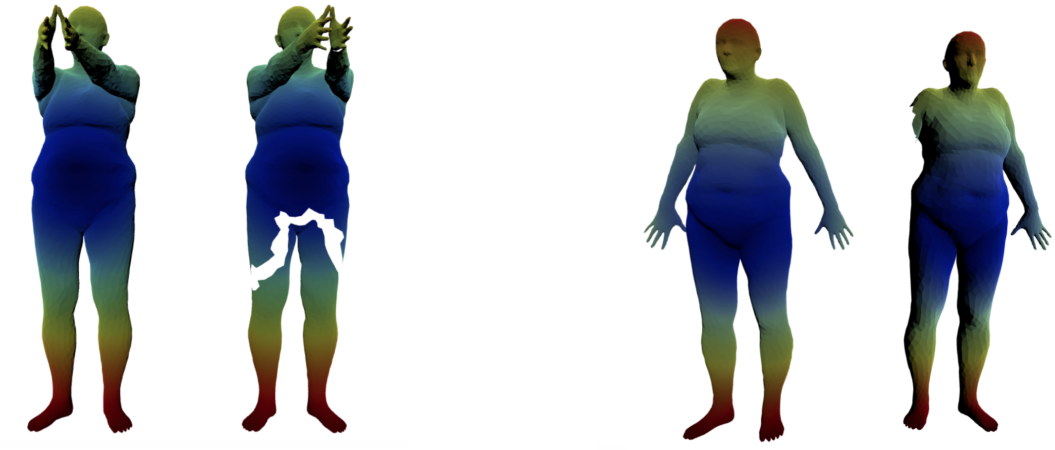


(a) (b)

Figure 4.2.: Quantitative comparison measured in-terms of Geodesic distortion as function of percentage of correspondence on Left : FAUST-Original and Right: FAUST-remesh dataset.

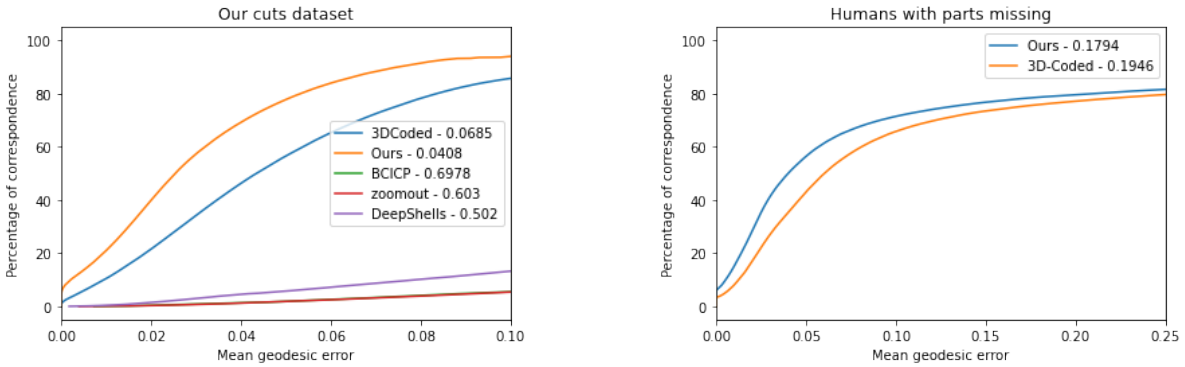
4.4.3. Generalisation to cuts and holes

We further explore the generalisability of our approach with respect to benign partiality and noise. To test this, we consider the aforementioned test set of 20 meshes and introduce partiality in the form of cuts and holes as visualised in Figure 4.3. Similar to the previous experiment, we train on the first 80 meshes of the original FAUST dataset and test the correspondence accuracy for matching between these partial shapes and part-to-whole. Quantitative comparison results are illustrated in Figure 4.4 and summarised in Table 4.1. For the part-to-whole setting, we only compare our method against 3D-Coded as other methods which are based on Functional Map, do not necessarily apply to this setting as the Laplacian [90] is only defined for parts and not the shape as a whole. Our method marginally performs better than 3D-Coded while both the template based methods do not deteriorate as significantly as others in the presence of strong noise. We show more qualitative examples of different settings in the Appendix section Figures A.6, A.7, A.8.



(a) (b)

Figure 4.3.: An example visualising our created part dataset from FAUST-Remesh dataset. Left : our dataset with random cuts, Right : Our dataset with holes.



(a) (b)

Figure 4.4.: Quantitative comparison measured in-terms of Geodesic distortion as function of percentage of correspondence between different methods on our dataset created from the test set of FAUST-remesh dataset. Left : Cuts dataset Right : Holes dataset.

Dataset	BCICP	ZoomOut	DeepShells	3D-Coded	Ours
FAUST	0.065	0.026	0.007	0.064	0.033
FAUST-Remesh	0.105	0.060	0.047	0.035	0.028
Cuts	0.698	0.603	0.502	0.068	0.048
Holes	-	-	-	0.195	0.179

Table 4.1.: Summary of mean geodesic error of different approaches on various dataset mentioned above.

4.4.4. Observation and Remarks

In this section we demonstrated that implicit latent representations are better suited for template based non-rigid 3D Shape correspondence than its counterpart extrinsic method, 3D-Coded. We remark that while our method is trained on the original FAUST dataset, it performs nearly twice better than the existing state-of-the-art method DeepShells on the FAUST-Remesh dataset. Furthermore, on our generated parts and cuts dataset based on FAUST-Remesh, our performs better than all the baseline. However, on the original FAUST dataset, ZoomOut and DeepShells show better performance than our method. We posit that simple mesh connectivity to be the reason for them to outperform us on the relatively simple FAUST dataset. While our method has shown signs of robustness on the FAUST dataset, there are currently some drawbacks which we address in the next section.

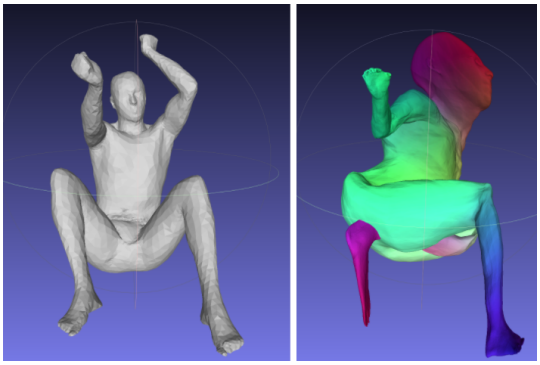
4.5. Drawbacks and Work in progress

While initial results of using an implicit latent representation looks encouraging on the FAUST dataset, however, our approach largely fails to generalise. In this section, we will discuss two such scenarios.

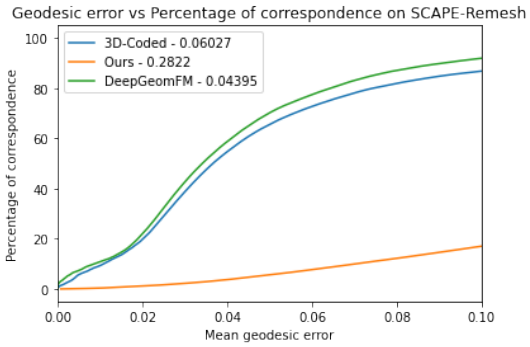
4.5.1. Adapting to unseen poses

Our approach largely fails to generalise to unseen poses at training time. In particular, the FAUST dataset contained poses in the test set which already was a part of the training set. We illustrate this with an example on the SCAPE dataset [110], which contains new unseen poses in the test set. We train our method and 3D-Coded on randomly sub-sampled 4k meshes from the SURREAL dataset [111, 88]. While the SURREAL dataset contains meshes that are in 1-1 correspondence and in distinct poses, it does not contain *exactly* same poses in its training and test set, in contrast to the FAUST dataset. Surprisingly, 3D-Coded demonstrates convincing quantitative results as shown in Figure 4.5 while our approach falls short of producing acceptable results. The reason for the non-scalability of our approach and its lack

of generalisation is something which we do not fully understand.



(a)

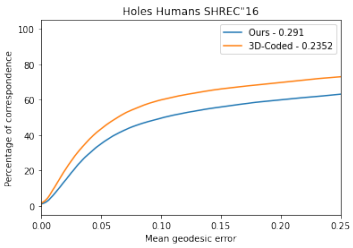


(b)

Figure 4.5.: Left : Original mesh and reconstructed mesh by our approach on SCAPE dataset. Right : Quantitative geodesic distortion error on SCAPE dataset.

In addition, while both 3D-Coded and our approach showed encouraging results on our proposed dataset, however, they fail to produce convincing results when the level of noise is significant such as the SHREC-16 dataset [112] as shown in Figure 4.6. While intuitively template based seems optimal for the task of partial setting, it is unclear as to why both the methods fail under severe partiality and is a point of interest to better understand the shortcomings.

(a)



(b)

Figure 4.6.: Visualising correspondence between whole-part matching setting from the holes challenge of SHREC-16 dataset. Left : Qualitative example and Right - Quantitative comparison between ours and 3D-Coded. Our approach performs inferior to 3D-Coded in contrast to previous settings.

5. Conclusion and Future Work

In this thesis, we have analysed several methods for 3D Shape reconstruction and used them as a main tool to perform 3D shape optimisation. We further demonstrated that such reconstruction based approaches can successfully be applied to the context of non-rigid Shape Correspondence. Our main contribution however was the Latent Shape Optimisation. We demonstrated that optimisation of different properties of a shape can be performed at the latent space given sufficient data annotated with respective properties.

While our approach is simple and effective, it opens many questions for the future research in 3D Computer Vision. In particular, which other subfields of Shape Analysis can benefit from the Deep Implicit Neural Networks? Can we improve the performance of a Network outside of generative modelling by optimising latent vectors through backpropagation? While these questions in itself are interesting, it would also be a momentous task to solve the dependency of a non-differentiable iso-surface extraction to obtain visual feedback of the reconstruction algorithm. In all our experimental settings discussed in this report, a latent vector corresponds to a shape which is implicitly defined and can only be visualised upon applying Marching Cubes which is a non-differentiable algorithm. It would be interesting to have a computationally feasible alternative wherein, we can have a visual feedback and apply a differentiable supervision upon the implicitly defined shape.

Finally, we hope that works described in this thesis can be successfully applied to real-world settings and have meaningful practical applications. It would be very interesting and exciting to see if our work in shape optimisation actually contributes to lesser plastic consumption.

A. Additional Figures

A.1. Comparison of Reconstruction Methods

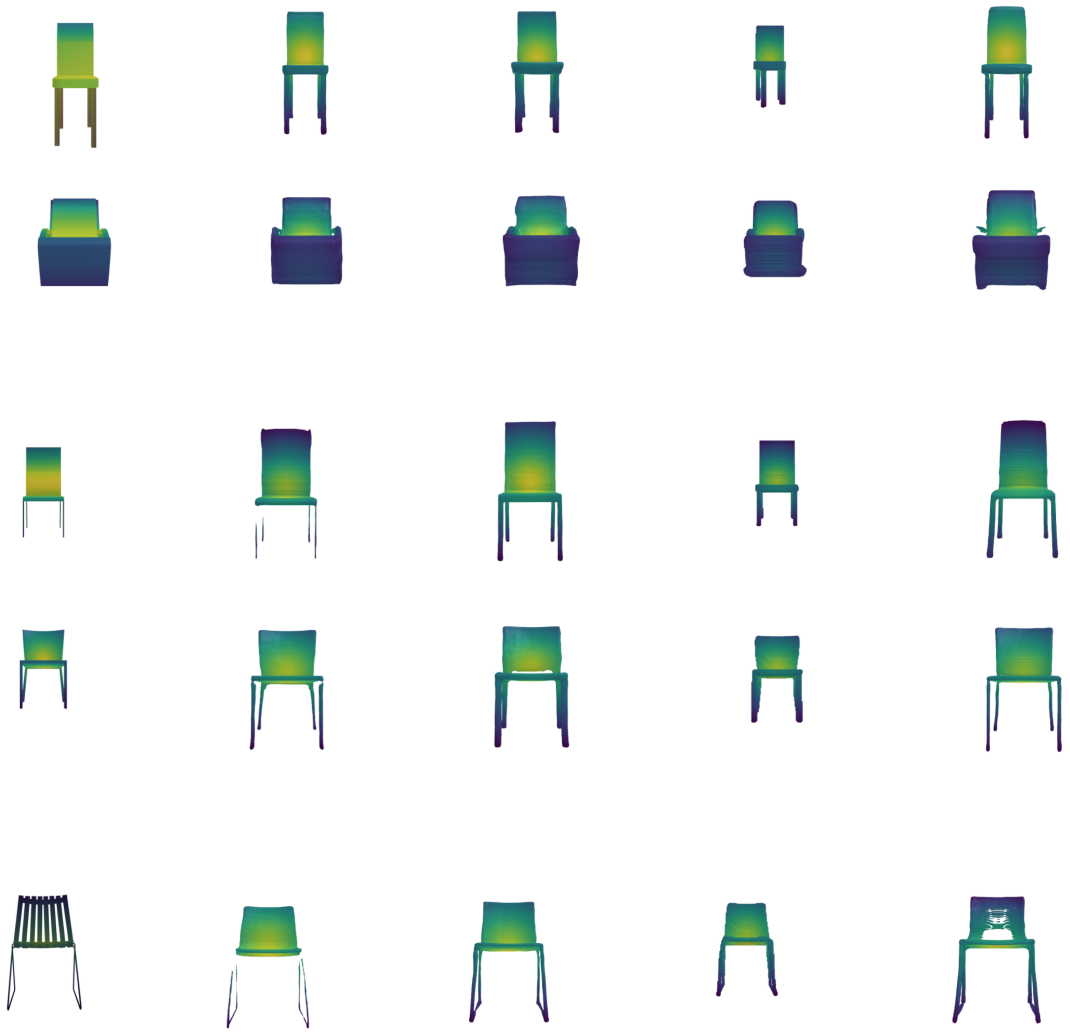


Figure A.1.: Comparison between various Deep Implicit 3D reconstruction methods on more examples from chair class of ShapeNetV2 dataset. Left to Right : Original mesh, DeepSDF, IM-Net, PQ-Net and DualSDF.

A.2. Comparison of High-Frequency Reconstruction Methods

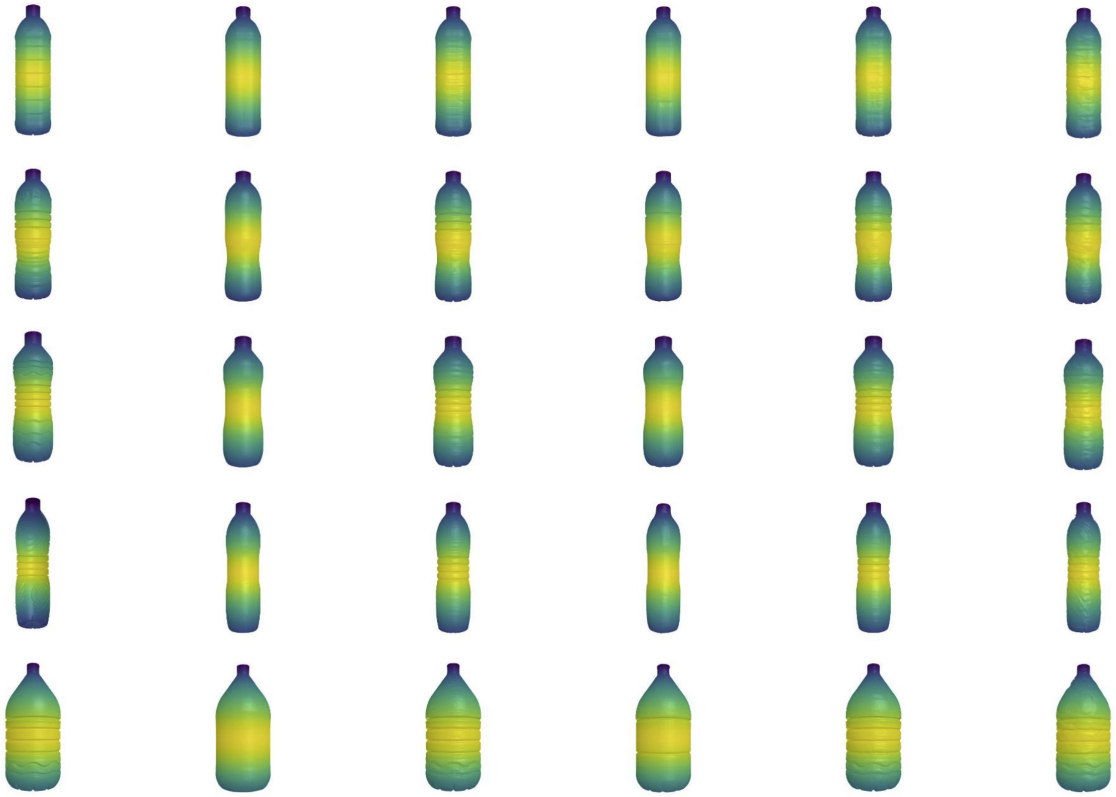


Figure A.2.: Comparison between various high-frequency reconstruction method on the Bottles dataset. Left to Right : Original, DeepSDF, DeepSDF+FFM, CSDF, CSDF+FFM, Modulated Network.

Figure A.3.: Comparison between various high-frequency reconstruction method on FAUST dataset. Left to Right : Original, DeepSDF, DeepSDF+FFM, CSDF, CSDF+FFM, Modulated Network.

A.3. More Latent Space Optimisation Visualisation

Figure A.4.: Qualitative comparison of different variants of LSO. Left to Right : A-LSO, L-LSO, C-LSO NN, C-LSO (k=4 neighbours).

A.4. More Non-rigid Shape Correspondence Visualisation

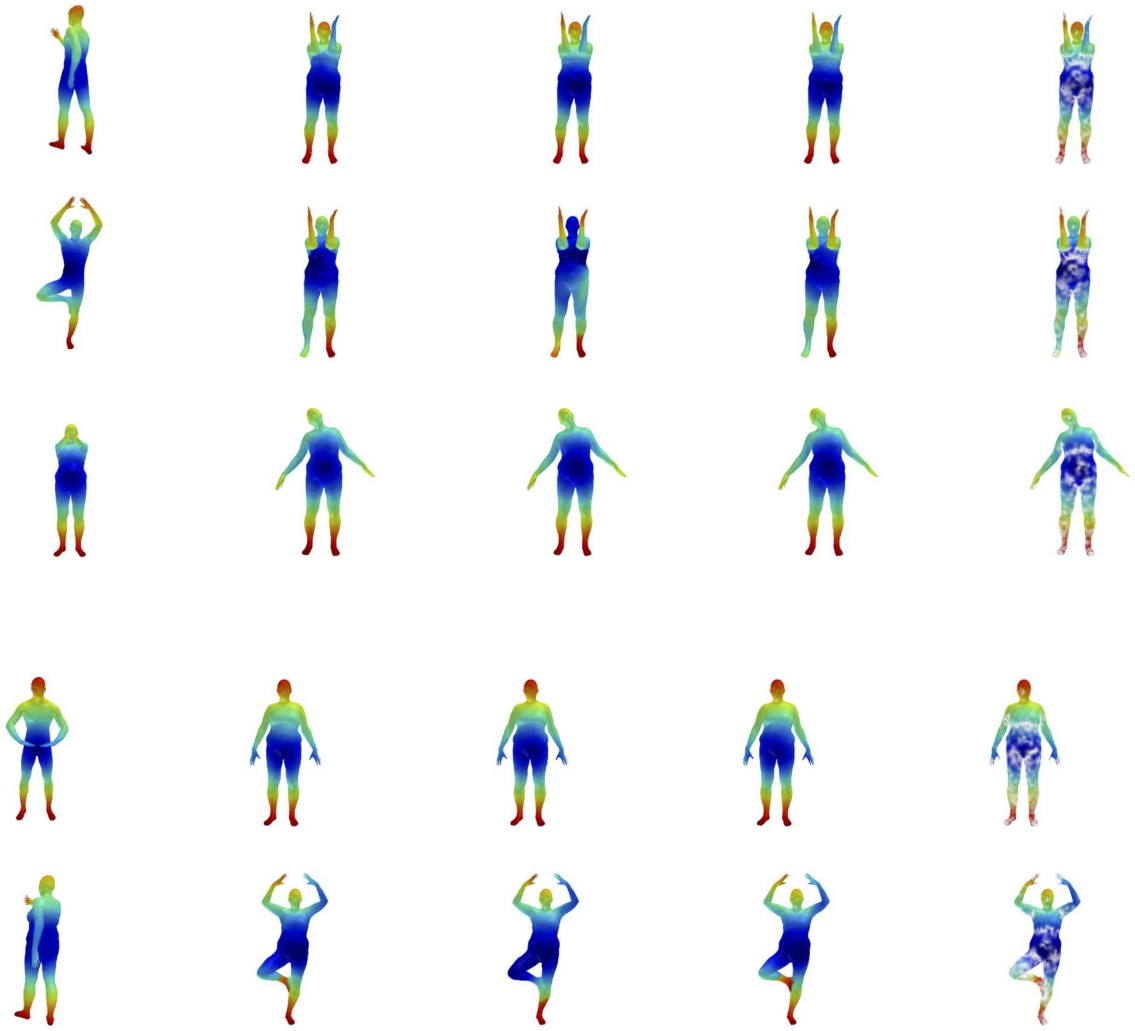


Figure A.5.: More qualitative comparison between different methods for non-rigid Shape Correspondence. Left to Right : Source mesh, Target mesh with correspondence color-coded by 1) 3D-Coded, 2) DeepShells, 3) Ours and 4) Ground Truth

A.5. More Non-rigid Partial Shape Correspondence Visualisation

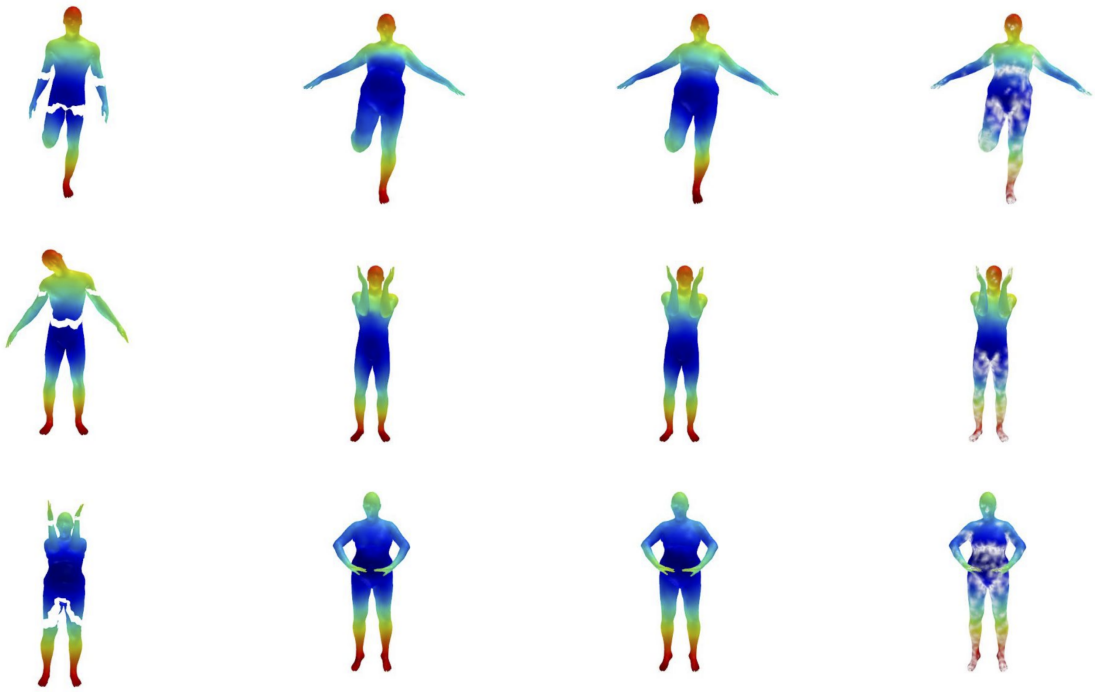


Figure A.6.: Qualitative comparison between different methods for part to whole matching. Left to Right : Source mesh, Target mesh with correspondence color-coded by 1) 3D-Coded, 2) DeepShells, 3) Ours and 4) Ground Truth

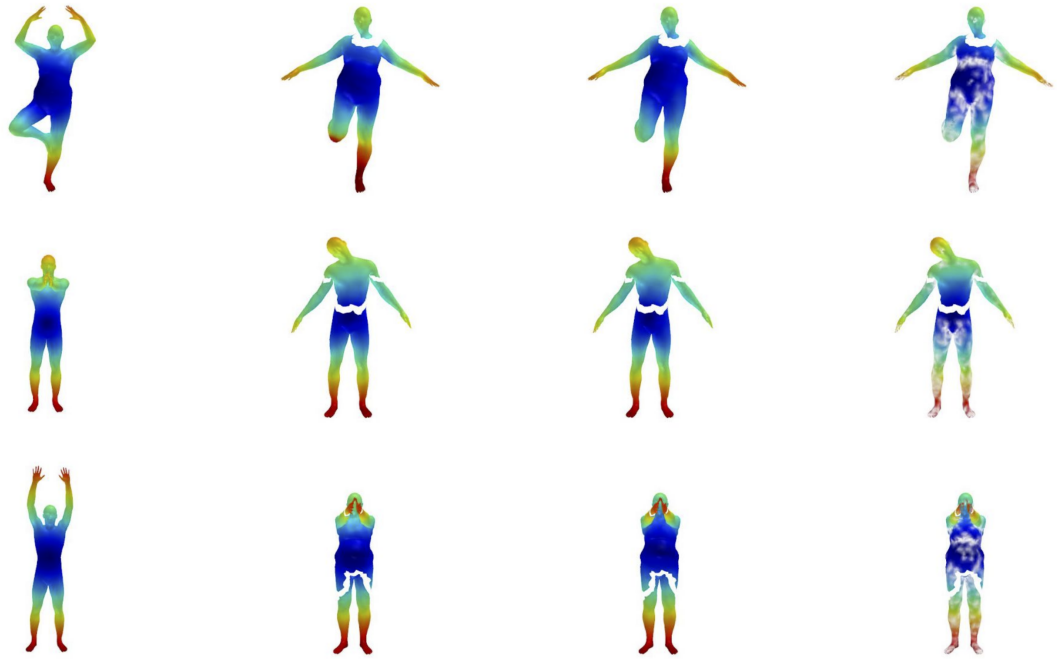


Figure A.7.: Qualitative comparison between different methods for whole to part matching. Left to Right : Source mesh, Target mesh with correspondence color-coded by 1) 3D-Coded, 2) DeepShells, 3) Ours and 4) Ground Truth

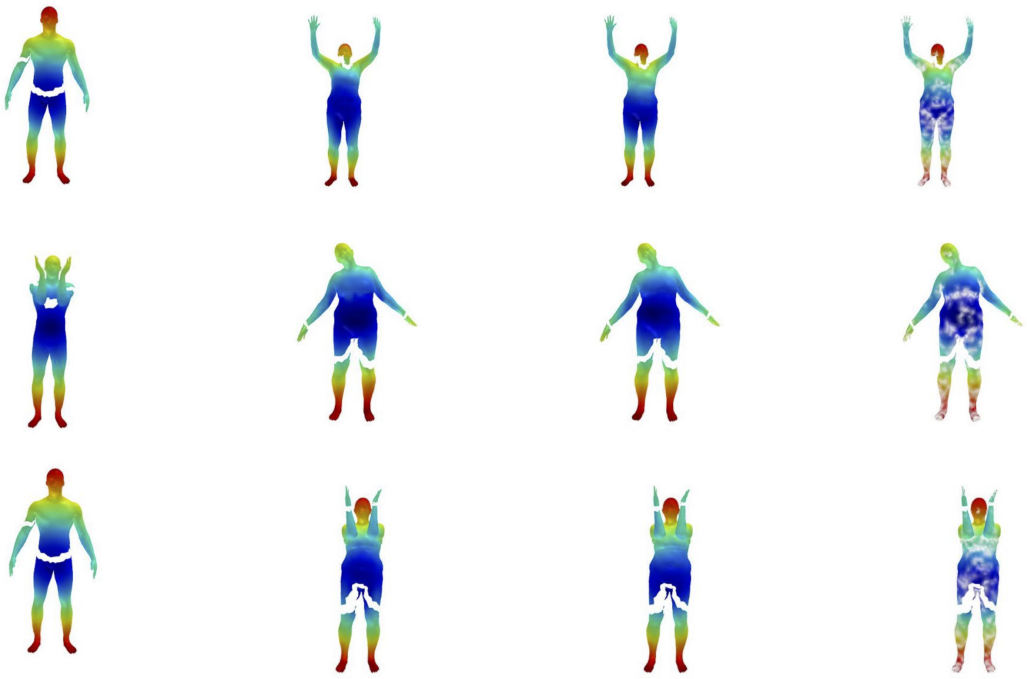


Figure A.8.: Qualitative comparison between different methods for part to part matching. Left to Right : Source mesh, Target mesh with correspondence color-coded by 1) 3D-Coded, 2) DeepShells, 3) Ours and 4) Ground Truth

List of Figures

List of Tables

Bibliography

[1] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. *DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation*. 2019. arXiv: 1901.05103 [cs.CV].

[2] R. Wu, Y. Zhuang, K. Xu, H. Zhang, and B. Chen. *PQ-NET: A Generative Part Seq2Seq Network for 3D Shapes*. 2020. arXiv: 1911.10949 [cs.CV].

[3] Z. Hao, H. Averbuch-Elor, N. Snavely, and S. Belongie. *DualSDF: Semantic Shape Manipulation using a Two-Level Representation*. 2020. arXiv: 2004.02869 [cs.CV].

[4] M. A. Uy, V. G. Kim, M. Sung, N. Aigerman, S. Chaudhuri, and L. Guibas. "Joint Learning of 3D Shape Retrieval and Deformation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[5] M. Mezghanni, M. Boulkenafed, A. Lieutier, and M. Ovsjanikov. "Physically-Aware Generative Network for 3D Shape Modeling". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 9330–9341.

[6] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. "Functional Maps: A Flexible Representation of Maps between Shapes". In: *ACM Trans. Graph.* 31.4 (July 2012). ISSN: 0730-0301. DOI: 10.1145/2185520.2185526. URL: https://doi.org/10.1145/2185520.2185526.

[7] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov. *ZoomOut: Spectral Upsampling for Efficient Shape Correspondence*. 2019. arXiv: 1904.07865 [cs.GR].

[8] J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. "Continuous and Orientation-Preserving Correspondences via Functional Maps". In: *ACM Trans. Graph.* 37.6 (Dec. 2018). ISSN: 0730-0301. DOI: 10.1145/3272127.3275040. URL: https://doi.org/10.1145/3272127.3275040.

[9] M. Eisenberger, Z. Lähner, and D. Cremers. *Smooth Shells: Multi-Scale Shape Registration with Functional Maps*. 2019. arXiv: 1905.12512 [cs.CV].

[10] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. *Deep Shells: Unsupervised Shape Correspondence with Optimal Transport*. 2020. arXiv: 2010.15261 [cs.CV].

[11] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. *Learning Representations and Generative Models for 3D Point Clouds*. 2018. URL: https://openreview.net/forum?id=BJInEZsTb.

[12] A. Hertz, R. Hanocka, R. Giryes, and D. Cohen-Or. *PointGMM: a Neural GMM Network for Point Clouds*. 2020. arXiv: 2003.13326 [cs.LG].

[13] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan. "PointFlow: 3D Point Cloud Generation With Continuous Normalizing Flows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[14] M. Arjovsky, S. Chintala, and L. Bottou. *Wasserstein GAN*. 2017. arXiv: 1701.07875 [stat.ML].

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. *Improved Training of Wasserstein GANs*. 2017. arXiv: 1704.00028 [cs.LG].

[16] D. J. Rezende and S. Mohamed. *Variational Inference with Normalizing Flows*. 2016. arXiv: 1505.05770 [stat.ML].

[17] D. Rempe, T. Birdal, Y. Zhao, Z. Gojcic, S. Sridhar, and L. J. Guibas. "CaSPR: Learning Canonical Spatiotemporal Point Cloud Representations". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[18] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. "3D ShapeNets: A deep representation for volumetric shapes". In: June 2015, pp. 1912–1920. DOI: 10.1109/CVPR.2015.7298801.

[19] Y. Guan, T. Jahan, and O. van Kaick. "Generalized Autoencoder for Volumetric Shape Generation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 1082–1088. DOI: 10.1109/CVPRW50498.2020.00142.

[20] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling". In: *CoRR* abs/1610.07584 (2016). arXiv: 1610.07584. URL: http://arxiv.org/abs/1610.07584.

[21] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. "O-CNN: Octree-Based Convolutional Neural Networks for 3D Shape Analysis". In: *ACM Trans. Graph.* 36.4 (July 2017). ISSN: 0730-0301. DOI: 10.1145/3072959.3073608. URL: https://doi.org/10.1145/3072959.3073608.

[22] G. Riegler, A. O. Ulusoy, and A. Geiger. *OctNet: Learning Deep 3D Representations at High Resolutions*. 2017. arXiv: 1611.05009 [cs.CV].

[23] M. Tatarchenko, A. Dosovitskiy, and T. Brox. "Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs". In: *IEEE International Conference on Computer Vision (ICCV)*. 2017. URL: http://lmb.informatik.uni-freiburg.de/Publications/2017/TDB17b.

[24] S. Cheng, M. M. Bronstein, Y. Zhou, I. Kotsia, M. Pantic, and S. Zafeiriou. "MeshGAN: Non-linear 3D Morphable Models of Faces". In: *CoRR* abs/1903.10384 (2019). arXiv: 1903.10384. URL: http://arxiv.org/abs/1903.10384.

[25] Q. Tan, L. Gao, Y.-K. Lai, and S. Xia. *Variational Autoencoders for Deforming 3D Mesh Models*. 2018. arXiv: 1709.04307 [cs.GR].

[26] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia. *Deformable Shape Completion with Graph Convolutional Autoencoders*. 2018. arXiv: 1712.00268 [cs.CV].

[27] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. "MeshCNN: A Network with an Edge". In: *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301. DOI: 10.1145/3306346.3322959. URL: https://doi.org/10.1145/3306346.3322959.

[28] L. Gao, J. Yang, T. Wu, Y.-J. Yuan, H. Fu, Y.-K. Lai, and H. Zhang. *SDM-NET: Deep Generative Network for Structured Deformable Mesh*. 2019. arXiv: 1908.04520 [cs.GR].

[29] C. Nash and C. K. I. Williams. "The shape variational autoencoder: A deep generative model of part-segmented 3D objects". In: *Computer Graphics Forum* 36.5 (Aug. 2017), pp. 1–12. DOI: 10.1111/cgf.13240. URL: https://doi.org/10.1111/cgf.13240.

[30] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. *AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation*. 2018. arXiv: 1802.05384 [cs.CV].

[31] M. Defferrard, X. Bresson, and P. Vandergheynst. *Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering*. 2017. arXiv: 1606.09375 [cs.LG].

[32] T. N. Kipf and M. Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG].

[33] W. E. Lorensen and H. E. Cline. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm". In: *SIGGRAPH Comput. Graph.* 21.4 (Aug. 1987), pp. 163–169. ISSN: 0097-8930. DOI: 10.1145/37402.37422. URL: https://doi.org/10.1145/37402.37422.

[34] A. Zadeh, Y.-C. Lim, P. P. Liang, and L.-P. Morency. *Variational Auto-Decoder: A Method for Neural Generative Modeling from Incomplete Data*. 2021. arXiv: 1903.00840 [cs.LG].

[35] Z. Chen and H. Zhang. "Learning Implicit Fields for Generative Shape Modeling". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[36] I. Sutskever, O. Vinyals, and Q. V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL].

[37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE].

[38] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. *ShapeNet: An Information-Rich 3D Model Repository*. Tech. rep. arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[39] A. Zadeh, Y.-C. Lim, P. P. Liang, and L.-P. Morency. *Variational Auto-Decoder: A Method for Neural Generative Modeling from Incomplete Data*. 2021. arXiv: 1903.00840 [cs.LG].

[40] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. Courville. *On the Spectral Bias of Neural Networks*. 2019. arXiv: 1806.08734 [stat.ML].

[41] A. Bietti and J. Mairal. *On the Inductive Bias of Neural Tangent Kernels*. 2019. arXiv: 1905.12173 [stat.ML].

[42] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*. 2020. arXiv: 2006.10739 [cs.CV].

[43] A. Rahimi and B. Recht. "Random Features for Large-Scale Kernel Machines". In: *Proceedings of the 20th International Conference on Neural Information Processing Systems*. NIPS'07. Vancouver, British Columbia, Canada: Curran Associates Inc., 2007, pp. 1177–1184. ISBN: 9781605603520.

[44] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. *Implicit Neural Representations with Periodic Activation Functions*. 2020. arXiv: 2006.09661 [cs.CV].

[45] V. Sitzmann, M. Zollhöfer, and G. Wetzstein. *Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations*. 2020. arXiv: 1906.01618 [cs.CV].

[46] Y. Deng, J. Yang, and X. Tong. *Deformed Implicit Field: Modeling 3D Shapes with Learned Dense Correspondence*. 2021. arXiv: 2011.13650 [cs.CV].

[47] Y. Duan, H. Zhu, H. Wang, L. Yi, R. Nevatia, and L. J. Guibas. *Curriculum DeepSDF*. 2020. arXiv: 2003.08593 [cs.CV].

[48] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. "Curriculum learning". In: *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*. Ed. by A. P. Danyluk, L. Bottou, and M. L. Littman. Vol. 382. ACM International Conference Proceeding Series. ACM, 2009, pp. 41–48. DOI: 10.1145/1553374.1553380. URL: https://doi.org/10.1145/1553374.1553380.

[49] F. Bogo, J. Romero, M. Loper, and M. J. Black. "FAUST: Dataset and evaluation for 3D mesh registration". In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ, USA: IEEE, June 2014.

[50] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[51] G. Allaire, E. Bonnetier, G. Francfort, and F. Jouve. "Shape optimization by the homogenization method". In: *Numerische Mathematik* 76.1 (Mar. 1997), pp. 27–68. DOI: 10.1007/s002110050253. URL: https://doi.org/10.1007/s002110050253.

[52] R. Prévost, E. Whiting, S. Lefebvre, and O. Sorkine-Hornung. "Make It Stand: Balancing Shapes for 3D Fabrication". In: *ACM Trans. Graph.* 32.4 (July 2013). ISSN: 0730-0301. DOI: 10.1145/2461912.2461957. URL: https://doi.org/10.1145/2461912.2461957.

[53] H. Zhao, W. Xu, K. Zhou, Y. Yang, X. Jin, and H. Wu. "Stress-Constrained Thickness Optimization for Shell Object Fabrication". In: *Computer Graphics Forum* 36.6 (Sept. 2016), pp. 368–380. DOI: 10.1111/cgf.12986. URL: https://doi.org/10.1111/cgf.12986.

[54] M. Bächer, B. Bickel, E. Whiting, and O. Sorkine-Hornung. "Spin-It: Optimizing Moment of Inertia for Spinnable Objects". In: *Commun. ACM* 60.8 (July 2017), pp. 92–99. ISSN: 0001-0782. DOI: 10.1145/3068766. URL: https://doi.org/10.1145/3068766.

[55] N. Umetani, T. Igarashi, and N. J. Mitra. "Guided Exploration of Physically Valid Shapes for Furniture Design". In: *ACM Transactions on Graphics* 31.4 (2012), 86:1–86:11.

[56] T. Liu, A. Hertzmann, W. Li, and T. Funkhouser. "Style Compatibility for 3D Furniture Models". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 34.4 (Aug. 2015).

[57] A. Schulz, A. Shamir, D. I. W. Levin, P. Sitthi-amorn, and W. Matusik. "Design and Fabrication by Example". In: *ACM Trans. Graph.* 33.4 (July 2014). ISSN: 0730-0301. DOI: 10.1145/2601097.2601127. URL: https://doi.org/10.1145/2601097.2601127.

[58] G. Xu, X. Liang, S. Yao, D. Chen, and Z. Li. "Multi-Objective Aerodynamic Optimization of the Streamlined Shape of High-Speed Trains Based on the Kriging Model". In: *PLOS ONE* 12.1 (Jan. 2017). Ed. by W.-B. Du, e0170803. DOI: 10.1371/journal.pone.0170803. URL: https://doi.org/10.1371/journal.pone.0170803.

[59] P. Baqué, E. Remelli, F. Fleuret, and P. Fua. *Geodesic Convolutional Shape Optimization*. 2018. arXiv: 1802.04016 [cs.CE].

[60] R. J. Preen and L. Bull. "Toward the Coevolution of Novel Vertical-Axis Wind Turbines". In: *IEEE Transactions on Evolutionary Computation* 19.2 (Apr. 2015), pp. 284–294. DOI: 10.1109/tevc.2014.2316199. URL: https://doi.org/10.1109/tevc.2014.2316199.

[61] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin. "Modeling by Example". In: *ACM SIGGRAPH 2004 Papers*. SIGGRAPH '04. Los Angeles, California: Association for Computing Machinery, 2004, pp. 652–663. ISBN: 9781450378239. DOI: 10.1145/1186562.1015775. URL: https://doi.org/10.1145/1186562.1015775.

[62] A. Lundberg, P. Hamlin, D. Shankar, A. Broniewicz, T. Walker, and C. Landström. "Automated Aerodynamic Vehicle Shape Optimization Using Neural Networks and Evolutionary Optimization". In: *SAE International Journal of Passenger Cars - Mechanical Systems* 8.1 (Apr. 2015), pp. 242–251. DOI: 10.4271/2015-01-1548. URL: https://doi.org/10.4271/2015-01-1548.

[63] N. Thuerey, K. Weißenow, L. Prantl, and X. Hu. "Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows". In: *AIAA Journal* 58.1 (Jan. 2020), pp. 25–36. DOI: 10.2514/1.j058291. URL: https://doi.org/10.2514/1.j058291.

[64] A. Poulenard, P. Skraba, and M. Ovsjanikov. "Topological Function Optimization for Continuous Shape Matching". In: *Computer Graphics Forum* 37.5 (Aug. 2018), pp. 13–25. DOI: 10.1111/cgf.13487. URL: https://doi.org/10.1111/cgf.13487.

[65] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. *Geodesic convolutional neural networks on Riemannian manifolds*. 2018. arXiv: 1501.06297 [cs.CV].

[66] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. 2017. arXiv: 1612.00593 [cs.CV].

[67] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. 2017. arXiv: 1706.02413 [cs.CV].

[68] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. *Spatial Transformer Networks*. 2016. arXiv: 1506.02025 [cs.CV].

[69] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. *KPConv: Flexible and Deformable Convolution for Point Clouds*. 2019. arXiv: 1904.08889 [cs.CV].

[70] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. *Dynamic Graph CNN for Learning on Point Clouds*. 2019. arXiv: 1801.07829 [cs.CV].

[71] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov. *DiffusionNet: Discretization Agnostic Learning on Surfaces*. 2021. arXiv: 2012.00888 [cs.CV].

[72] L. Shapira, A. Shamir, and D. Cohen-Or. "Image Appearance Exploration by Model-Based Navigation". In: *Computer Graphics Forum* (2009). ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2009.01403.x.

[73] M. Ovsjanikov, W. Li, L. Guibas, and N. J. Mitra. "Exploration of Continuous Variability in Collections of 3D Shapes". In: *ACM Transactions on Graphics* 30.4 (2011), to appear.

[74] J. O. Talton, D. Gibson, L. Yang, P. Hanrahan, and V. Koltun. "Exploratory Modeling with Collaborative Design Spaces". In: *ACM Trans. Graph.* 28.5 (Dec. 2009), pp. 1–10. ISSN: 0730-0301. DOI: 10.1145/1618452.1618513. URL: https://doi.org/10.1145/1618452.1618513.

[75] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun. "A Probabilistic Model for Component-Based Shape Synthesis". In: *ACM Trans. Graph.* 31.4 (July 2012). ISSN: 0730-0301. DOI: 10.1145/2185520.2185551. URL: https://doi.org/10.1145/2185520.2185551.

[76] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan. "Symmetry in 3D Geometry: Extraction and Applications". In: *Computer Graphics Forum* 32.6 (Feb. 2013), pp. 1–23. DOI: 10.1111/cgf.12010. URL: https://doi.org/10.1111/cgf.12010.

[77] A. Tevs, Q. Huang, M. Wand, H.-P. Seidel, and L. Guibas. "Relating Shapes via Geometric Symmetries and Regularities". In: *ACM Trans. Graph.* 33.4 (July 2014). ISSN: 0730-0301. DOI: 10.1145/2601097.2601220. URL: https://doi.org/10.1145/2601097.2601220.

[78] M. A. Uy, J. Huang, M. Sung, T. Birdal, and L. Guibas. *Deformation-Aware 3D Model Embedding and Retrieval*. 2020. arXiv: 2004.01228 [cs.CV].

[79] M. E. Yumer and N. J. Mitra. "Learning Semantic Deformation Flows with 3D Convolutional Networks". In: *European Conference on Computer Vision (ECCV 2016)*. Springer. 2016, pp. -.

[80] W. Yifan, N. Aigerman, V. G. Kim, S. Chaudhuri, and O. Sorkine-Hornung. *Neural Cages for Detail-Preserving 3D Deformations*. 2020. arXiv: 1912.06395 [cs.GR].

[81] W. Wang, D. Ceylan, R. Mech, and U. Neumann. *3DN: 3D Deformation Network*. 2019. arXiv: 1903.03322 [cs.CV].

[82] M. Savva, A. X. Chang, and P. Hanrahan. "Semantically-Enriched 3D Models for Common-sense Knowledge". In: *CVPR 2015 Workshop on Functionality, Physics, Intentionality and Causality* (2015).

[83] H. Hoppe. "New Quadric Metric for Simplifiying Meshes with Appearance Attributes". In: *Proceedings of the Conference on Visualization '99: Celebrating Ten Years*. VIS '99. San Francisco, California, USA: IEEE Computer Society Press, 1999, pp. 59–66. ISBN: 078035897X.

[84] M. Aubry, U. Schlickewei, and D. Cremers. "The wave kernel signature: A quantum mechanical approach to shape analysis". In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 2011, pp. 1626–1633. DOI: 10.1109/ICCVW.2011.6130444.

[85] J. Sun, M. Ovsjanikov, and L. Guibas. "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion". In: *Computer Graphics Forum* 28.5 (July 2009), pp. 1383–1392. DOI: 10.1111/j.1467-8659.2009.01515.x. URL: https://doi.org/10.1111/j.1467-8659.2009.01515.x.

[86] M. Andreux, E. Rodola, M. Aubry, and D. Cremers. "Anisotropic Laplace-Beltrami Operators for Shape Analysis". In: *Sixth Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*. 2014.

[87] R. Huang, J. Ren, P. Wonka, and M. Ovsjanikov. "Consistent ZoomOut: Efficient Spectral Map Synchronization". In: *Symposium of Geometry Processing 2020*. Vol. 39. 5. Utrecht, Netherlands, July 2020, pp. 265–278. DOI: 10.1111/cgf.14084. URL: https://hal.inria.fr/hal-02953729.

[88] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. "Shape correspondences from learnt template-based parametrization". In: *15th European Conference on Computer Vision ECCV 2018*. Munich, Germany, Sept. 2018. URL: https://hal.archives-ouvertes.fr/hal-01830474.

[89] T. Deprelle, T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. *Learning elementary structures for 3D shape generation and matching*. 2019. arXiv: 1908.04725 [cs.CV].

[90] S. Rosenberg. *The Laplacian on a Riemannian Manifold: An Introduction to Analysis on Manifolds*. London Mathematical Society Student Texts. Cambridge University Press, 1997. DOI: 10.1017/CBO9780511623783.

[91] A. Johnson. "Spin-Images: A Representation for 3-D Surface Matching". PhD thesis. Pittsburgh, PA: Carnegie Mellon University, Aug. 1997.

[92] S. Belongie, J. Malik, and J. Puzicha. "Shape Context: A New Descriptor for Shape Matching and Object Recognition". In: *Advances in Neural Information Processing Systems*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. MIT Press, 2001. URL: https://proceedings.neurips.cc/paper/2000/file/c44799b04a1c72e3c8593a53e8000c78-Paper.pdf.

[93] R. Gal, A. Shamir, and D. Cohen-Or. "Pose-Oblivious Shape Signature". In: *IEEE Transactions on Visualization and Computer Graphics* 13.2 (Mar. 2007), pp. 261–271. DOI: 10.1109/tvcg.2007.45. URL: https://doi.org/10.1109/tvcg.2007.45.

[94] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas. "One Point Isometric Matching with the Heat Kernel". In: *Computer Graphics Forum* 29.5 (Sept. 2010), pp. 1555–1564. DOI: 10.1111/j.1467-8659.2010.01764.x. URL: https://doi.org/10.1111/j.1467-8659.2010.01764.x.

[95] H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman. "Point Registration via Efficient Convex Relaxation". In: *ACM Trans. Graph.* 35.4 (July 2016). ISSN: 0730-0301. DOI: 10.1145/2897824.2925913. URL: https://doi.org/10.1145/2897824.2925913.

[96] Y. Soliman, D. Slepčev, and K. Crane. "Optimal Cone Singularities for Conformal Flattening". In: *ACM Trans. Graph.* 37.4 (2018).

[97] K. Crane, U. Pinkall, and P. Schröder. "Spin Transformations of Discrete Surfaces". In: *ACM Trans. Graph.* 30.4 (July 2011). ISSN: 0730-0301. DOI: 10.1145/2010324.1964999. URL: https://doi.org/10.1145/2010324.1964999.

[98] N. Donati, A. Sharma, and M. Ovsjanikov. *Deep Geometric Functional Maps: Robust Feature Learning for Shape Correspondence*. 2020. arXiv: 2003.14286 [stat.ML].

[99] J.-M. Roufosse, A. Sharma, and M. Ovsjanikov. *Unsupervised Deep Learning for Structured Shape Matching*. 2019. arXiv: 1812.03794 [cs.GR].

[100] R. Marin, M.-J. Rakotosaona, S. Melzi, and M. Ovsjanikov. "Correspondence learning via linearly-invariant embedding". In: *NeurIPS*. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/11953163dd7fb12669b41a48f78a29b6-Abstract.html.

[101] T. Caissard, D. Coeurjolly, J.-O. Lachaud, and T. Roussillon. "Heat Kernel Laplace-Beltrami Operator on Digital Surfaces". In: *Discrete Geometry for Computer Imagery*. Springer International Publishing, 2017, pp. 241–253. DOI: 10.1007/978-3-319-66272-5_20. URL: https://doi.org/10.1007/978-3-319-66272-5_20.

[102] O. Litany, T. Remez, E. Rodolà, A. M. Bronstein, and M. M. Bronstein. *Deep Functional Maps: Structured Prediction for Dense Shape Correspondence*. 2017. arXiv: 1704.08686 [cs.CV].

[103] S. Salti, F. Tombari, and L. D. Stefano. "SHOT: Unique signatures of histograms for surface and texture description". In: *Computer Vision and Image Understanding* 125 (Aug. 2014), pp. 251–264. DOI: 10.1016/j.cviu.2014.04.011. URL: https://doi.org/10.1016/j.cviu.2014.04.011.

[104] S. Zuffi and M. J. Black. "The stitched puppet: A graphical model of 3D human shape and pose". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3537–3546. DOI: 10.1109/CVPR.2015.7298976.

[105] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. "SMPL: A Skinned Multi-Person Linear Model". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6 (Oct. 2015), 248:1–248:16.

[106] Z. Zheng, T. Yu, Q. Dai, and Y. Liu. *Deep Implicit Templates for 3D Shape Representation*. 2021. arXiv: 2011.14565 [cs.CV].

[107] Y. Deng, J. Yang, and X. Tong. "Deformed Implicit Field: Modeling 3D Shapes with Learned Dense Correspondence". In: *IEEE Computer Vision and Pattern Recognition*. 2021.

[108] V. G. Kim, Y. Lipman, and T. Funkhouser. "Blended Intrinsic Maps". In: *ACM Trans. Graph.* 30.4 (July 2011). ISSN: 0730-0301. DOI: 10.1145/2010324.1964974. URL: https://doi.org/10.1145/2010324.1964974.

[109] D. Nogneng and M. Ovsjanikov. "Informative Descriptor Preservation via Commutativity for Shape Matching". In: *Computer Graphics Forum* (2017). ISSN: 1467-8659. DOI: 10.1111/cgf.13124.

[110] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. "SCAPE: Shape Completion and Animation of People". In: *ACM Trans. Graph.* 24.3 (July 2005), pp. 408–416. ISSN: 0730-0301. DOI: 10.1145/1073204.1073207. URL: https://doi.org/10.1145/1073204.1073207.

[111] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. "Learning from Synthetic Humans". In: *CVPR*. 2017.

[112] L. Cosmo, E. Rodolà, M. M. Bronstein, A. Torsello, D. Cremers, and Y. Sahillioglu. "Partial Matching of Deformable Shapes". In: *Eurographics Workshop on 3D Object Retrieval*. Ed. by A. Ferreira, A. Giachetti, and D. Giorgi. The Eurographics Association, 2016. ISBN: 978-3-03868-004-8. DOI: 10.2312/3dor.20161089.