UNIVERSITÉ PARIS-EST

Learning 3D Generation and Matching

Thibault GROUEIX

A doctoral thesis in the domain of automated signal and image processing supervised by **Mathieu AUBRY**

Submitted to **École Doctorale Paris-Est**

Mathématiques et Sciences et Technologies de l'Information et de la Communication.

Presented on Thursday, 22nd of October 2020 to a committee consisting of:

Mathieu AUBRY	École des Ponts ParisTech	Supervisor
Renaud MARLET	École des Ponts ParisTech	Supervisor
Niloy MITRA	University College London	Reviewer
Pascal FUA	Ecole polytechnique fédérale de Lausanne	Reviewer
Cordelia SCHMID	INRIA	Examiner
Matthew FISHER	Adobe Research	Examiner

École des Ponts ParisTech
LIGM-IMAGINE
6, Av Blaise Pascal - Cité Descartes
Champs-sur-Marne
77455 Marne-la-Vallée cedex 2
France

Université Paris-Est Marne-la-Vallée
École Doctorale Paris-Est MSTIC
Département Études Doctorales
6, Av Blaise Pascal - Cité Descartes
Champs-sur-Marne
77454 Marne-la-Vallée cedex 2
France

Ce manuscrit est dédié a mon grand-père Guy Groueix.

Abstract

The goal of this thesis is to develop deep learning approaches to model and analyse 3D shapes. Progress in this field could democratize artistic creation of 3D assets which currently requires time and expert skills with technical software. We focus on the design of deep learning solutions for two particular tasks, key to many 3D modeling applications: single-view reconstruction and shape matching.

A single-view reconstruction (SVR) method takes as input a single image and predicts a 3D model of the physical world which produced that image. SVR dates back to the early days of computer vision. In particular, in the 1960s, Lawrence G. Roberts proposed to align simple 3D primitives to an input image making the assumption that the physical world is made of simple geometric shapes like cuboids. Another approach proposed by Berthold Horn in the 1970s is to decompose the input image in intrinsic images and use those to predict the depth of every input pixel. Since several configurations of shapes, texture and illumination can explain the same image, both approaches need to make assumptions on the distribution of textures and 3D shapes to resolve the ambiguity. In this thesis, we learn these assumptions from large-scale datasets instead of manually designing them. Learning SVR also allows to reconstruct complete 3D models, including parts which are not visible in the input image.

Shape matching aims at finding correspondences between 3D objects. Solving this task requires both a local and global understanding of 3D shapes which is hard to achieve. We propose to train neural networks on large-scale datasets to solve this task and capture knowledge implicitly through their internal parameters. Shape matching supports many 3D modeling applications such as attribute transfer, automatic rigging for animation, or mesh editing.

The first technical contribution of this thesis is a new parametric representation of 3D surfaces which we model using neural networks. The choice of data representation is a critical aspect of any 3D reconstruction algorithm. Until recently, most of the approaches in deep 3D model generation were predicting volumetric voxel grids or point clouds, which are discrete representations. Instead, we present an alternative approach that predicts a parametric surface deformation *i.e.* a mapping from a template to a target geometry. To demonstrate the benefits of such a representation, we train a deep encoder-decoder for single-view reconstruction using our new representation. Our approach, dubbed AtlasNet, is the first deep single-view reconstruction approach able to reconstruct meshes from images without relying on an independent post-processing. And it can perform such a reconstruction at arbitrary resolution without memory issues. A more detailed analysis of AtlasNet reveals it also generalizes better to categories it has not been trained on than other deep 3D generation approaches.

Our second main contribution is a novel shape matching approach based purely on reconstruction via deformations. We show that the quality of the shape reconstructions is critical to obtain good correspondences, and therefore introduce a test-time optimization scheme to refine the learned deformations. For humans and other deformable shape categories deviating by a near-isometry, our approach can leverage a shape template and isometric regularization of the surface deformations. As category exhibiting non-isometric variations, such as chairs, do not have a clear template, we also learn how to deform any shape into any other and leverage cycle-consistency constraints to learn meaningful correspondences. Our matching-by-reconstruction strategy operates directly on point clouds, is robust to many types of perturbations, and outperformed the state of the art by 15% on dense matching of real human scans.
Keywords: deep learning, surface generation, single-view reconstruction, shape matching

Résumé

L'objectif de cette thèse est de développer des approches d'apprentissage profond pour modéliser et analyser les formes 3D. Les progrès dans ce domaine pourraient démocratiser la création artistique de modèles 3D, actuellement réservée à quelques experts du domaine et couteuse en temps. En particulier, nous nous concentrons sur deux tâches clefs pour la modélisation 3D : reconstruire un modèle 3D à partir d'une seule image et mettre des modèles 3D en correspondance.

Une méthode de reconstruction 3D à partir d'une seule image (SVR) est un algorithme qui prend comme entrée une seule image et prédit un modèle 3D du monde physique qui a produit cette image. Ce problème remonte aux premiers jours de la vision par ordinateur. Étant donné que plusieurs configurations de formes, de textures et d'éclairage peuvent expliquer la même image il faut formuler des hypothèses sur la distribution des textures et des formes 3D pour résoudre cette ambiguïté. Dans cette thèse, nous apprenons ces hypothèses directement à partir de grandes bases de données, au lieu de les concevoir manuellement ad hoc. Les méthodes d'apprentissage pour la SVR nous permettent aussi d'effectuer une reconstruction complète et réaliste de l'objet, y compris des parties qui ne sont pas visibles dans l'image d'entrée.

La mise en correspondance de formes vise à établir des correspondances entre des objets 3D. Résoudre cette tâche nécessite à la fois une compréhension locale et globale des formes 3D qui est difficile à obtenir. Pour cela, nous proposons d'entrainer des réseaux neuronaux sur de grands jeux de données pour apprendre ces connaissances implicitement. La mise en correspondance de formes a de nombreuses applications en modélisation 3D telles que le transfert d'attribut, le gréement automatique pour l'animation ou l'édition de maillage.

La première contribution technique de cette thèse est une nouvelle représentation paramétrique des surfaces 3D, que nous modélisons avec des réseaux neuronaux. Le choix de la représentation des données est un aspect critique de tout algorithme de reconstruction 3D. Jusqu'à récemment, la plupart des approches profondes en génération 3D prédisaient des grilles volumétriques de voxel ou des nuages de points, qui sont des représentations discrètes. Au lieu de cela, nous présentons une approche qui prédit une déformation paramétrique de surface, c'est-à-dire une déformation d'un modèle source vers une forme objectif. Pour démontrer les avantages de cette nouvelle représentation, nous l'utilisons pour la reconstruction 3D à partir d'une seule image. Notre approche, baptisée AtlasNet, est la première approche profonde de SVR capable de reconstruire des maillages à partir d'images sans s'appuyer sur un post-traitement, et peut le faire à une résolution arbitraire sans problèmes de mémoire. Une analyse plus détaillée d'AtlasNet révèle qu'il généralise également mieux que les autres approches par apprentissage aux catégories sur lesquelles il n'a pas été entrainé.

Notre deuxième contribution est une nouvelle approche de correspondance de formes entièrement basée sur des reconstructions par déformation de surface. Nous montrons que la qualité des reconstructions 3D est essentielle pour obtenir de bonnes correspondances. Nous introduisons donc une optimisation au moment de l'inférence pour affiner les déformations apprises. Pour les humains et d'autres catégories de formes déformables qui diffèrent d'une quasi-isométrie, notre approche peut tirer parti d'un modèle de catégorie et d'une régularisation des déformations vers l'isométrie. Comme les catégories présentant des variations non isométriques, telles que les chaises, n'ont pas de modèle clair, nous apprenons à déformer n'importe quelle forme en n'importe quelle autre et tirons parti des contraintes de cohérence du cycle pour apprendre des correspondances qui respectent la sémantique des objets. Notre approche de correspondance de formes fonctionne directement sur les nuages de points, elle est robuste à de nombreux types de perturbations et a surpassé l'état de l'art de 15% sur des scans d'humains réels.
Mots clés : apprentissage profond, génération de surface, reconstruction à partir d'une seule image, correspondance de formes

Acknowledgements

Je remercie chaleureusement les membres de mon jury ainsi que les rapporteurs de la thèse. Leurs pertinents conseils contribuent à ces pages.

Un étudiant en thèse est un bateau d'explorateurs. On commence jeune embarcation téméraire en quête de découvertes lointaines. On se jette sur des mers obscures en se demandant si les planches sont prêtes pour un si long voyage. Parfois bringuebalé sur des eaux tumultueuses, parfois immobile sur des mers d'huile. Les premières tempêtes sont là, et le bateau et son équipage tiennent bon. Puis ce sont les îlots que l'on entre-aperçoit. Certain sont riches et fertiles, on y trouve même des pistes vers de nouveaux trésors. D'autres sont des culs-de-sac où Calypso endort le voyageur dans de fausses espérances. Les années passent et les cales se remplissent de découvertes extravagantes. Il est déjà temps de finir le voyage et le bateau rentre à son port alourdi de trésors exotiques. Au pays, les Hommes ne retiendront que les cales et ses trésors, qui sont livrés à travers ces pages, mais le bateau sait ce qu'il doit à son équipage, et quel trésor fut le voyage.

D'abord son capitaine, Mathieu. Travailler sous ta direction fut un honneur. Merci de m'avoir transmis ta passion pour la recherche et les découvertes. Je retiens que tu m'as soutenu par tous les ciels, clairs comme orageux. Ta disponibilité, tes idées audacieuses, et ta confiance nous on fait franchir de nombreuses eaux. Le mou idoine que tu laissais sur la barre de direction du navire lui permet aujourd'hui de tenter une navigation autonome.
Ensuite, les seconds du navire, Mat, Vova, Bryan. Je vous remercie pour deux étés dans les eaux californiennes extrêmement riches en découvertes, et trois années de sereine navigation sur toutes les mers.
Je remercie le chef de la Garde et quartier-maître, Renaud, pour m'avoir efficacement défendu afin que je puisse orienter mes voiles vers l'Amérique, alors que l'embargo semblait possible, et m'avoir aidé à de nombreuses étapes.
Je remercie Théo, d'abord jeune mousse, puis navire à son tour. Nous naviguâmes ensemble jusqu'aux illustres mers du grand Ouest Canadiens. Merci d'avoir été assez fou pour pousser avec moi la génération de chaises encore plus loin. Merci à Tom d'avoir partagé avec moi un dernier sprint final.
Je remercie tous les membres de l'équipage scientifique qui imaginent des horizons nouveaux. Chacun contribue au foisonnement des idées mais c'est surtout l'énergie et la bienveillance quotidienne de chacun qui bonifie l'âme du bateau avec le temps.

Parmi ceux-là, je remercie le salsero de la rumba, de la guaracha et du guaguenco : mon ami Pierre-Alain Langlois. Pour son aide scientifique, l'ambiance qu'il met sur le pont, et tous les souvenirs construits ensemble.
Je remercie les vieux loups de mer qui m'ont accueilli en 2016 : Martin, Spyros, Marina, Laura, Francisco, Shell, Praveer et Sergey. Merci Francisco pour ton rôle temporaire de timonier, ton aide m'a mis sur des courants favorables dans les premiers mois de ma thèse. Martin, fin navigateur, qui a affûté mes azimuts le dimanche dans les forêts d'Ile-de-France. Quelle joie de mettre le cap sur l'Adriatique pour le mariage de Spyros, le responsable des quarts nocturnes. Merci aussi aux jeunes mousses qui nous ont rejoint, vite aguerris à l'air marin. La Brésilienne est une technique rarement utile en navigation mais merci à François, Abdou et Tom de m'y avoir entraîné. Merci Yang et Xi d'avoir insisté pour que nous campions au cours de notre road trip post-CVPR. Merci aux acolytes du Descartes, qui a toujours su ragaillardir nos cerveaux moulus par le vent : Othman, Benjamin, Xuchong, et tous les amis d'Imagine attablés pendant que nos grand-voiles (GPUs) rugissaient sous l'effort.
Enfin, je remercie tous les passagers qui ont trouvé leur place dans ce navire étrange. Chers colocs de la CC, vous m'avez soutenu dans les semaines pré-deadline et avaient même accepté ma chaise en papier-mâché comme décoration du salon. Theresa, merci d'avoir été la petite goutte de folie dans mes nuits de recherches tardives. Merci à Etienne qui court décidément très vite (mais en rond sur le pont), à Gabriel, ses (bras)tasses et ses techniques de nœuds marins toujours TTA, Maud le charpentier du bateau pour ses aménagements astucieux, Matthieu et son sprint final de Ljubljana, et Lucie d'avoir été le cœur de ce foyer chaleureux.
Je remercie aussi les explorateurs d'un autre genre qui m'ont accompagné dans une entreprise similaire. Claire voyageait dans l'espace quand Paulin était au jeu. Antoine parcourait les glaciers rocheux tandis que Corentin se renforçait.
Je remercie ceux qui m'ont insufflé l'envie de ce voyage exploratoire : Malik et Gilles. L'impulsion donnée en 2015, alors que je n'étais qu'un canot sanglé à leur vaisseau, m'a laissé un goût de reviens-y-voir.
Je remercie l'armateur du navire, l'École des Ponts et le Labex Bezout pour leur financement.
Je remercie les parents Doumergue ainsi que leurs pioupious de m'avoir ouvert leur port(e) quand naviguer dehors devint interdit et qu'il fallut remiser génois et spi pendant plusieurs mois.
Maman, Papa, Émilie, nous naviguons sur des mers différentes mais le bois de nos planches vient du même arbre. Merci d'exister et d'être ma famille.
Maud, merci de me soutenir, de gonfler ma voile, et de m'aimer. Si j'ai laissé quelques cheveux dans ces cinq articles, je n'ai pu le faire sereinement que grâce à toi.

Contents

Chapter 1

Introduction

1.1 Goals

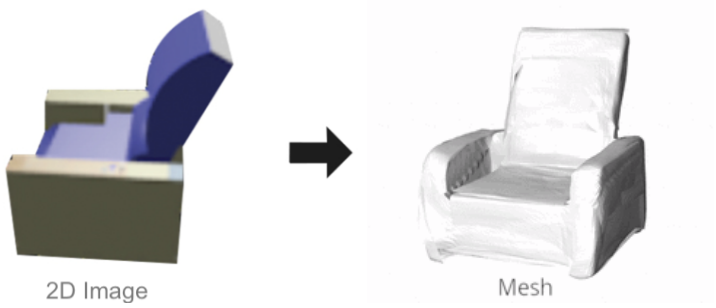
The goal of this thesis is to develop deep learning methods to model and analyse 3D shapes with a focus on two 3D computer vision tasks: (i) 3D shape reconstruction from a single image (ii) and correspondence estimation between 3D models. These two tasks are illustrated in Figure 1.1.

Single-view reconstruction aims at reconstructing the full 3D geometry of an object or a scene from an image. For example, in Figure 1.1, a 3D mesh is generated from a single 2D RGB image. Note that we try to hallucinate the unseen parts in the image, which makes the problem very hard. In contrast with early approaches based on optimization and hand-crafted priors, the goal of this thesis is to develop a deep learning solution for this task and learn data-driven priors from large-scale datasets. In particular, we focus on the problem of finding a 3D data representation that is compatible with a deep learning approach, and can capture 3D shape information at a high resolution. In chapter 3, we introduce the first deep method that reconstructs a mesh out of a single image.

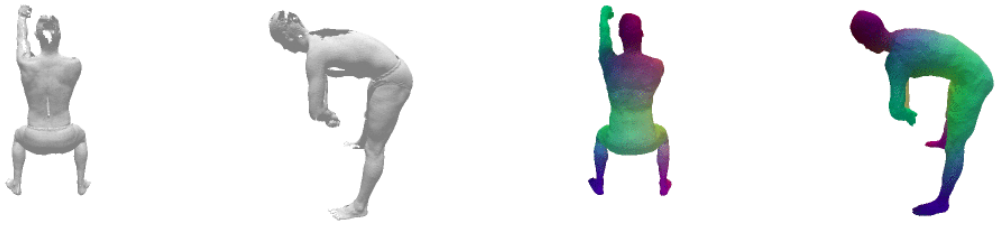
3D shape matching aims at estimating point-to-point correspondences between two 3D models. It is a long-standing challenge of computational geometry. Shape matching has a wide range of applications such as 3D scan alignment, attribute transfer between shapes, and shape interpolation. In contrast to classical methods that solve this problem independently on each pair of shapes, the goal of this thesis is to train deep learning solutions on large-scale datasets to solve the shape matching problem jointly on many pairs. In chapter 4, we introduce a new approach to solve this tasks by reconstructing shapes via their deformation from a common template. Figure 1.1 shows a visualization of the shape matching problem, tackled with our approach on a human pair. Our approach improves state-of-the-art performances on standards benchmarks. In chapter 5, we lift the requirement for a common template and extend the method to all categories of objects, even those exhibiting high intra-class variations like the chair category.

1.2 Motivations

Single-image reconstruction and 3D shape matching are motivated by a wide range of industrial applications spanning 3D modeling, augmented reality and virtual reality, scene understanding, image understanding, and robotics. Figure 1.2 and figure 1.3 illustrates these applications.



(a) Single-view shape reconstruction: Given an input image, our approach presented in Chapter 3 creates a parametric 3D polygonal mesh. Note how the unseen parts are correctly hallucinated.



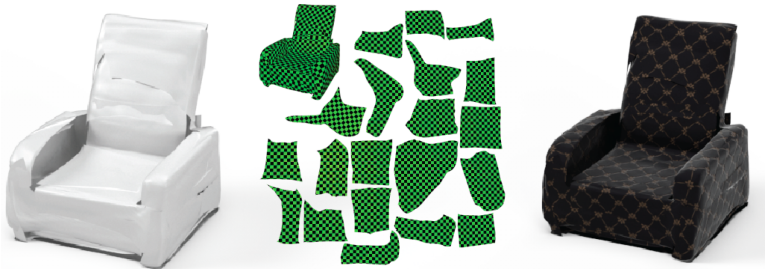
(b) Shape Matching: Given two input shapes without correspondences (left) , our approach presented in Chapter 4 establishes dense correspondences between them (right). Correspondences are suggested by color.

Figure 1.1 The two different tasks addressed in this thesis : single-view shape reconstruction and matching. Animated figures, best seen in Acrobat Reader.

3D modeling. Our main motivation for tackling shape matching and single-view reconstruction is to support digital artists with next-generation tools to author 3D content. 3D creation is hard, time-consuming and requires a lot of expertise. Similar to a painting that requires many layers of paint, many sub-tasks must be completed to build a 3D model. Take for instance 3D character design. The artist starts with coarse geometry creation, then takes many local passes to refine it with wrinkles, veins, skin hair, skin color and finally ends with rigging. Throughout the creation process, 3D artists rely on their skills and a palette of local editing tools, applying one stroke at a time. Instead, efficient shape matching and shape generation algorithms would enable higher level controls in order to author 3D assets.

Ideally, developing deep learning approaches for shape matching and single-view reconstruction leads to three high-level tools illustrated in Figure 1.2:

- **Warm-start modeling tools** that project coarse user input to the space of "plausible" 3D shapes. Figure 1.1 shows a single-view reconstruction example and Figure 1.2a shows an example of adding textures automatically to a model.



(a) Mesh parameterization. We establish continuous bijections between a 3D surface (left) and optimized planar patches (middle). This is particularly useful to apply texture on a 3D object (right).

(b) Signal transfer accross arbitrary shapes. We transfer a toy checkerboard signal from a source object (left) to a target untextured object by deforming the source in the target (middle). Colors suggest correspondences.

(c) Shape Morphing We interpolate between two real chairs, as a way to continuously explore the collection of chairs. Click here for the video.

Figure 1.2 Applications in this thesis useful for 3D modelling.

- **refinement tools** that support the user with high-level controls to customize existing shapes using attribute transfer based on correspondences. For instance, Figure 1.2b shows transfer of textures between two real shapes.
- **exploration tools** that enable 3D artists to search the space of realistic shapes a reconstruction algorithm can produce. See for instance in Figure 1.2c a shape interpolation.

Smart image editing. Some image editions, like viewpoint modification or object insertion/removal, requires a good 3D understanding of the scene depicted in the image. Indeed, during a small view-point modification, the 2D projection of a frontal object evolves differently from the 2D projection of a background object, which tend to vary less. It also involves difficult visibility issues : parts of the objects, originally unseen, appear in the new viewpoint, while others become occluded. Getting a plausible 3D representation from an image is thus a very useful proxy task for image editing since such a 3D model captures all information of depth and visibility from any viewpoint. For example, Mildenhall et al. (2020) propose an editing tool to change the camera view-point *a posteriori* and hallucinate unseen parts at a high resolution using 3D reconstruction techniques (see Figure 1.3b).

Shape generation and matching could in particular be used for:

- **Object manipulation,** to realistically rotate an object in an image by respecting illumination changes and resolving the new visibility issues.
- **Object insertion/removal,** to add or remove an object from an image by respecting illumination changes and resolving the new visibility issues.

Augmented Reality. Augmented Reality (AR) is an emerging field with huge prospective impact. One of its form boils down to building a digital 3D model of the world around us and "augmenting" it by inserting additional digital object (see Figure 1.3c). Key to this task is the ability to recreate a 3D model of the environment in real-time, with methods that are robust to the sensor's noise. In that regard, current results in shape generation are already promising and show that we can generate coarse 3D geometry from images in milliseconds Groueix et al. (2018a); Häne et al. (2017); Mescheder et al. (2019); Park et al. (2019a). Recently, Gkioxari et al. (2019) showcased a joint detection/reconstruction pipeline that reconstructs a full 3D scene with a separate 3D primitive for each object. Another important aspect of AR is the ability to change the appearance of existing object, by applying digital paint on a physical object. The key to achieve this is to put the reconstructed object in dense correspondence with another one and transfer appearance attributes between them. Generation and matching methods could be used to design:

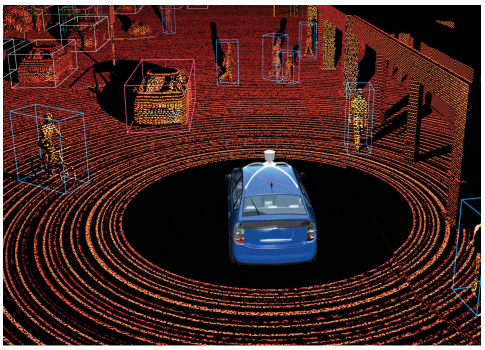
(a) 3D modelling: ZBRUSH gallery, *Ana de Armas.* High-quality 3D modelling takes time and expertise, for lack of high-level smart control tools.

(b) Smart Image Editing: Mildenhall et al. (2020) reconstruct a 3D scene from images. A user can then explore novel view-points *a posteriori*.

(c) Augmented Reality: Trying new furniture configurations at home before buying.[a]

[a] http://www.design-confidential.com/5037-2/

(d) Virtual Reality 3D reconstruction by CyArk prior to a 2016 damaging earthquake in Bagan, Myanmar. Built during the 10th century.



(e) Robotics: Application to autonomous driving. Ideally, all moving objects are detected by the 3D LIDAR. [a]

[a] https://www.popsci.com/cars/article/2013-09/google-self-driving-car/

(f) Smart humans controls: Using action detection, Ayotle turn physical objects into a tactile interfaces. [a]

[a] http://www.influencia.net/fr/actualites/in,innovations,anytouch-monde-devient-tactile,2660.html

Figure 1.3 Industrial applications of this thesis. Animated figure best viewed in Acrobat Reader.

- **A style modification tool** modifying the color, texture pattern of any furniture from a large online database of exemplar furniture's without modifying the geometry of the furniture. For instance in Figure 1.2a, a new texture is applied automatically on a 3D object.

- **A content modification tool** modifying the geometry of objects such as a piece of furniture using a large online database of exemplar furniture's without modifying its texture.

Virtual Reality. Creating fully immersive worlds has strong connections to the previously mentioned field of 3D modeling. One of the particular interest of 3D-generation-from-photographs methods in Virtual Reality (VR) is the visualization of historical scenes that no longer exists in the physical world. As shown in Figure 1.3d, in Bagan, Myanmar, the start-up CyArk used multi-view reconstruction (MVR) methods to make a digital 3D reconstruction of the site before its partial destruction in the wakes of a 2016 earth-quake. More broadly, 3D reconstructions methods (MVR and SVR) could help archive our previous architectural heritage and change the way we engage with historical data.

Robotics. To make robots grasp objects, Corona et al. (2020) propose to first estimate the 3D shape of an object, and then plan a grasping motion. Their approach use our 3D shape representation, presented in Chapter 3. This idea also can also work for autonomous cars, first reconstructing their environment from their sensors (LIDARs) and then making a driving decision (see Figure1.3e). Note that in this case, the robot is equipped with more than a single RGB sensor to reconstruct 3D shapes. In this thesis, we consider the single-image reconstruction scenario.

Smart human controls. Putting humans and objects in correspondences enables new applications in human/machine interaction, as shown in Figure 1.3f. Think for instance of a human clapping his hands to light a room. More broadly, any object, regardless of its size or its surface, could be made tactile and interactive[1]. Correspondence results for humans using a method presented in this are shown in Figure 1.1b, and detailed in Chapter 4.

[1] Ayotle

1.3 Approach and Context

Single image 3D reconstruction and shape matching are complex tasks: a solution requires the answer to thousands of sub-questions. Imagine trying to reconstruct a photographed room in 3D. *"Which objects is in the room?"*, *"how are they arranged spatially?"*, *"Which are visible parts and which parts must be hallucinated?"*, *"Which priors can I use to estimate the occluded parts?"*. Early approaches tried to handcraft explicit priors, and give explicit answers to all these questions.

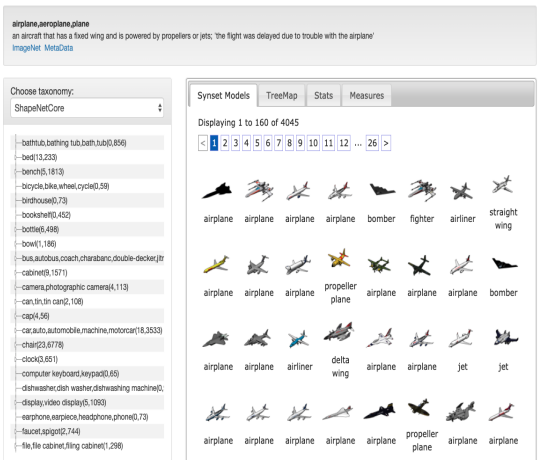
Instead, the central idea of this thesis is to learn priors for these tasks with deep learning from large-scale data collections.

We know from research in image generation that such an approach can be extremely powerful. Deep networks for image generation have progressed from low-resolution blurry image generation to high-quality detailed image generation Brock et al. (2019) in a short 5-year period. It has unleashed a tide of new digital art[2]. Similarly in 3D, we propose to exploit data to to learn priors for generation and matching and export the deep learning revolution to 3D creation.

This is a timely effort: more and more data is available that could allow to learn meaningful priors, and hardware/software solutions to process it keep improving. Scanning a 3D object has become as easy as opening a mobile app Kolev et al. (2014). Microsoft Kinects enable a user to easily capture an entire room Newcombe et al. (2011). The collective effort from industries and artists to build digital models of objects with Computer-aided Design has led to public libraries with millions of 3D objects: ShapeNet Chang et al. (2015) has over 3 millions models (see Figure1.4a), the ABC dataset Koch et al. (2019) has a million models. On the other hand, both academics and industrials like NVIDIA and Google have adapted Graphics Processing Units (GPU) from their original purpose to perform parallel tensor operations, which helps process this huge amount of data. In this thesis, GPUs are operated through the Pytorch library. Pytorch Paszke et al. (2019) is an open source machine learning framework without which we would still be implementing our first paper. The GPUs used for this thesis, NVIDIA's Titan X Pascal (see Figure 1.4b), have 3584 parallel workers operating at 1417MHz.

To sum up, we use deep learning to develop new methods for shape matching and single-view reconstruction. It can be done at this point in time because data and hardware have become massively available.

[2] e.g., Colie Wertz, *ship design*, Memo Akten, *Learning to see: Gloomy Sunday*



(a) Data: ShapeNet 3D object dataset. More than 51k annotated 3D objects, across 55 categories.

(b) Hardware: 4 NVIDIA Titan X Pascal GPU Hephaistos is not mentioned in the acknowledgement but he worked the hardest.

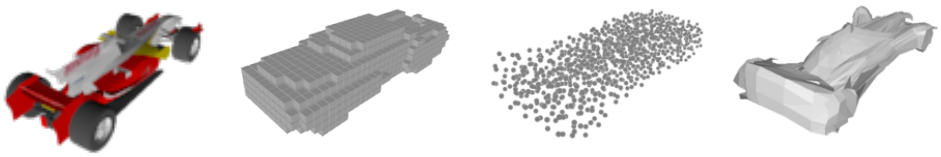
Figure 1.4 The backbones of this work: data and hardware.

1.4 Challenges

The first critical challenge we faced to develop deep learning approaches for 3D shape modelling was to design a suitable representation of 3D shapes. The second challenge we faced was to find ways to train neural networks with limited 3D annotations.

3D Data Representation for deep 3D generation. The choice of 3D data representation is a determining factor in most 3D approaches. A specific constraint to deep learning approaches is that the representation must also be a differentiable function. For instance, though meshes are widely used to represent 3D shapes, they are hard to generate with neural networks because properties such as mesh connectivity and number of points are discete. A standard representation for 3D generation and analysis has yet to be determined. A good representation should be a differentiable function, that can model fine-geometric details with reasonnable memory consumption.

Figure 1.5 compares three types of 3D data representation used in deep approaches for the task of single-image reconstruction. An appealing representation is discetized volumetric voxel grids since their are the natural extension of pixels in 3D. This allows to easily extend the successful 2D operators like convolution to 3D. However, this representation is memory-hungry and cannot capture fine details with current hardware constraints Choy et al. (2016). Another popular approach is to generate point clouds with neural networks which is memory efficient



(a) Input Image **(b)** Voxels **(c)** Point Cloud **(d)** Mesh

Figure 1.5 Challenge: 3D Data Representation. Comparison of three types of 3D data representation on the task of single-view reconstruction. From a 2D RGB image (a), 3D-R2N2 Choy et al. (2016) reconstructs a voxel-based 3D model (b), PointSetGen Fan et al. (2017) a point cloud based 3D model (c), and our AtlasNet a triangular mesh (d).

since it models the surface of 3D shape instead of their volume. Point clouds however lack the connectivity between the points which is useful in many downstream applications and makes them hard to visualize.

3D Annotations. Deep neural networks training requires a form of supervisory signal. Unfortunately, while 3D data is widely available, an important challenge for shape matching is that 3D correspondences annotations are scarce, expensive and not always well-defined. The task of single-image reconstruction also lacks large-scale datasets with dense 2D/3D annotations.

An ideal dataset for correspondences would have a large number of scans with varying shapes and pose. However, manual labeling is extremely expensive for the correspondence problem. A typical scan such as those used in Chapter 4's benchmark has about 10^5 points. Annotating correspondences points by points a single scan thus takes 28h at the impossible speed of 1 point / second.

Moreover, correspondence annotations are not always well defined. For a human scan, non-salient points on the skin are for instance hard to track with high precision through different poses. This problem is even more clear for categories of object exhibiting high-intra class variability like the "chair" category. Consider the two chairs in Figure 1.6: where does a point on the armrest of the deck chair map on the one-legged ball chair?

To generalize across many categories, we need to develop training strategies to learn generation and matching without point-to-point or pixel-to-point annotations. This is a hard and important problem: how to teach correspondences to a neural net without showing it correspondences?

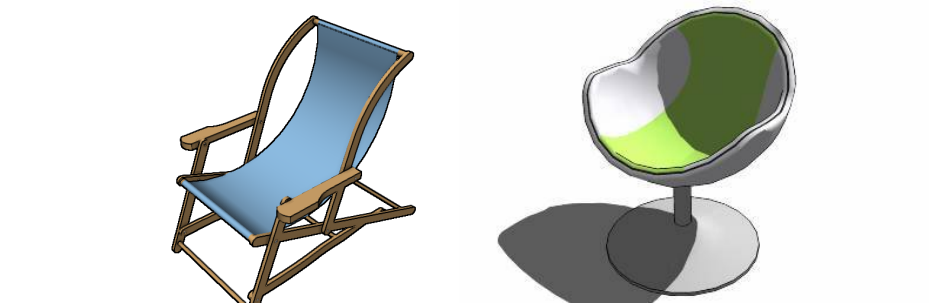


Figure 1.6 Challenge: 3D correspondence annotations. Two 3d chairs from the ShapeNet Chang et al. (2015) dataset, used in this thesis. Shape matching annotations are not avaible for such data with high topological variations.

1.5 Contributions

To tackle these challenges, we present two main contributions.

A Deep parametric 3D surface representation. We introduce a new 3D model representation based on deep parametric surface deformations. The key idea is to predict, with a simple Multi-Layered Perceptron, continuous parametric functions that can deform a template surface into target surfaces. After training, the neural network encodes a parametric family of deformation functions, and each shape is represented by one of these deformations. This deep representation models the surface of 3D shape. It is memory efficient, and can model fine-grained details. We use our new representation to introduce the first deep learning approach to direct single-image mesh reconstruction and to improve the state-of-the-art in shape matching.

Shape matching by deep deformation. We relate shape matching with 3D reconstruction by an analysis-by-synthesis strategy. We propose to densely match 3D models by reconstructing them via deep deformations. Key to the success of our matching-by-deformation approach is to learn accurate and semantically meaningful deformations. In the absence of annotated data, we use an unsupervised reconstruction loss, the Chamfer distance, to learn accurate reconstruction. For shape categories that deviate by near-isometries such as human, our approach leverage a shape template and we regularize our deformation towards isometry. For other categories exhibiting high intra-class topological variability such as chairs, we learn deformations of any shape into any other and enforce cycle-consistency constraints to learn meaningful correspondences. Our approach operates directly on raw point clouds. It is robust to many types of perturbation and outperformed the state of the art by 15% on human scans.

1.6 Thesis outline

This thesis is organized as follows:

Chapter 2: Related Work. We start by providing an overview of prior methods performing single-image 3D reconstruction and shape matching. We first discuss classic approaches, then deep-approaches leveraging data collections, most related to this thesis.

Chapter 3: AtlasNet. This chapter introduces the first contribution of this thesis: our new 3D data representation, based on parametric surface deformations. We first explain how deep neural networks can parameterize surface deformations. We then give theoretical guaranties and explain how our new representation is related to the mathematical definition of a surface. We empirically compare our representation with other 3D representations on the task of single-view reconstruction on the ShapeNet benchmark Chang et al. (2015). In particular, we show that our new 3D representation can generalize better to categories it has not been trained on. We also provide results showing its potential for other applications, such as morphing, parametrization, super-resolution, matching, and co-segmentation.

Chapter 4: 3D-CODED. We develop the idea of modelling deformations with deep networks to design a new approach to shape matching. Our approach follows a analysis-by-synthesis strategy: we show how to estimate state-of-the-art dense correspondences between human scans by reconstructing them via deep deformation of a common template. We provide evidence that the quality of the reconstructions are critical to get accurate correspondences. Based on this insight, we introduce an optimization scheme to refine the deformations predicted by the neural network. We experimentally compare against other approaches on the FAUST benchmark Bogo et al. (2014) and show that our matching-by-reconstruction approach improves on state-of-the-art, and is robust to many types of perturbations.

Chapter 5: Cycle-Consistent Deformations. We present a template-free method to perform shape matching in diverse shape collections, in the absence of annotations. Our approach builds on the success of our matching-by-reconstruction strategy. In the absence of a common template for classes exhibiting a large degree of topological variations, we propose to learn deformations of any shape in any other. To learn semantically meaningful deformations, we propose to use cycle-consistency to define a notion of good correspondences in groups of objects and use it as a supervisory signal to train our networks. We experimentally compare against other approaches on the task of segmentation transfer

across shapes from ShapeNet. We show that our approach is competitive with state-of-the-art methods when annotated training data is readily available, but outperforms them by a large margin when not much data is annotated.

Chapter 6: Conclusion. This chapter reflects on the contributions of the thesis and suggests on directions of future work.

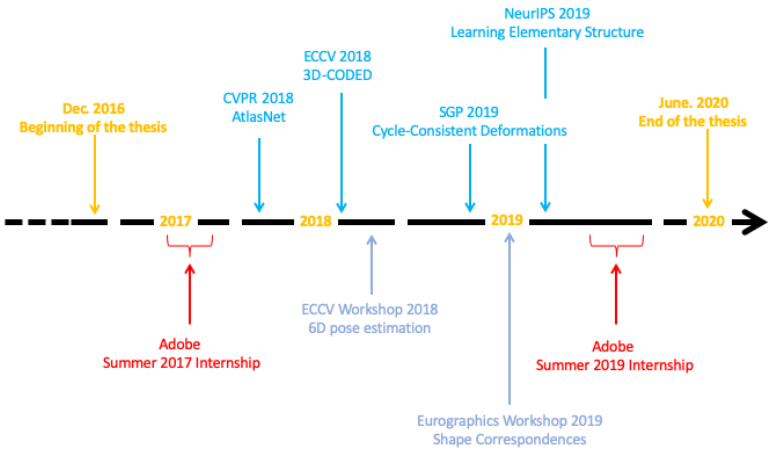


Figure 1.7 Thesis timeline from 2016 to 2020.

1.7 Publication List

Figure 1.7 summarises the highlights of this thesis. Three papers are presented in the manuscript.

- **Thibault Groueix**, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. (2018) AtlasNet: A Papier-Mache Approach to Learning 3D Surface Generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- **Thibault Groueix**, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. (2018) 3D-CODED : 3D Correspondences by Deep Deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- **Thibault Groueix**, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. (2019) Unsupervised cycle-consistent deformation for shape matching. *Computer Graphics Forum (SGP)*.

We open-sourced the code corresponding to the papers [3], incorporated AtlasNet in Kaolin [4], a python library designed to accelerate 3D Deep Learning Research by Jatavallabhula et al. (2019) and created webpages [5] for each project with additional visualizations of the results of the thesis. The webpages received overall around 12k unique visitors while the various codes on Github received a total of 800 stars and 110 forks. I received the best poster award for AtlasNet at the PAISS[6] summer school in 2018.

[3] https://github.com/ThibaultGROUEIX
[4] https://github.com/NVIDIAGameWorks/kaolin/
[5] http://imagine.enpc.fr/~groueixt/
[6] https://project.inria.fr/paiss/home-2018/

I also participated in two workshops Adelson and Pentland (1996); Hodan et al. (2018):

- Tomas Hodan and others. (2018) A Summary of the 4th International Workshop on Recovering 6D Object Pose. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- R. M. Dyke and others. (2019) Shape correspondence with isometric and non-isometric deformations. In *Eurographics Workshop on 3D Object Retrieval*

During my PhD, I also took part in two other projects which are not discussed in this manuscript Deprelle et al. (2019); Monnier et al. (2020):

- Theo Deprelle, **Thibault Groueix**, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. (2019) Learning elementary structures for 3D shape generation and matching. In *Advances in Neural Information Processing Systems (Neurips)*
- Tom Monnier, **Thibault Groueix**, and Mathieu Aubry. (2020) Deep Transformation-Invariant Clustering. In *Arxiv*.

I presented my work in several research teams, namely BAIR (Berkeley), MILA (LIP6 - Jussieu), Onera team DTIM, IGN Team Matis, LRI team Tau, Telecom ParisTech team 3D, Adobe Research, Huawei Research, and Naver Labs Europe.

Chapter 2

Related Work

This chapter briefly introduces deep learning, then reviews shape matching and single view reconstruction approaches related to our work.

2.1 Deep Learning

All the algorithms presented in this thesis are deep-learning-based. The thesis assumes that the reader is familiar with learning and in particular deep learning, and otherwise suggests Goodfellow et al. (2016) for a good introduction. We call our algorithms deep-learning-based because they incorporate these 4 key elements: a data collection, a neural network architecture, an energy function, and an optimization method to minimize the energy with respect to the parameters of the architecture.

- **Data collection.** It is the set of all the training samples. In this thesis, we rely on large public datasets like ShapeNet Chang et al. (2015) and SURREAL Varol et al. (2017). We use these datasets both in unsupervised frameworks, where we use the raw 3D data without any annotation, and in a supervised framework where we use annotations such as 2D renderings or correspondence annotations.

- **Neural network architecture.** It is a parametric family of differentiable functions. A common architecture, used in this thesis, is the Multi-layer-Perceptron (MPL) by Rosenblatt (1958). MPLs are stacks of linear and non-linear functions.

- **Energy function.** It is a differentiable function, chosen so that its minimum corresponds to the notion of "success" of the neural net output on the training samples.

- **Optimization method.** A global optimization is impossible because the function we want to minimize is in general non-convex. We thus rely on methods doing local optimization. Deterministic optimisation methods minimizing the energy function with gradient descent over the full data collection are usually infeasible due to its size. We thus rely on stochastic local optimization methods. In this thesis we use everywhere the Pytorch Paszke et al. (2019) implementation of the Adam optimizer Kingma and Ba (2014) which includes momentum and adaptative learning rates.

We draw our motivation to use deep learning from inspiring successes. Neural networks have famously outperformed humans at image classification He et al. (2016a); Huang et al. (2017); Krizhevsky et al. (2012), achieve high-quality image generation Goodfellow et al. (2014); Karras et al. (2019); Park et al. (2019b). They have also demonstrated super-human performances at games like Chess, Go Silver et al. (2016, 2018), and Starcraft Vinyals et al.

(2019). Applied on text data, neural networks translate text to speech van den Oord et al. (2016), speech to text Graves et al. (2013), and can translate content from one language to another Vaswani et al. (2017). Overall neural network have pushed the research boundaries in many fields. This thesis explores how they can be applied to shape matching and single-view reconstruction.

Autoencoders. Autoencoders are a particular type of neural networks used extensively in this thesis. Given an input sample, the encoder part of the autoencoder first compress it in a lower-dimensional latent code. From this latent code, the decoder part of the autoencoder attempts to reconstruct the input. By extension, we call encoder networks any neural network that learns a representation from an input signal (not only for reconstruction). In this thesis, we use a PointNet encoder from Qi et al. (2017a) to extract embeddings from point clouds, and a ResNet encoder from He et al. (2016b) to encode images. We call decoder network, any architecture that reconstructs a target from an implicit embedding. In Chapter 3, we learn a new decoder architecture which outputs deformed surfaces and compare against other types of decoder outputing point clouds and voxels.

2.2 Shape matching

Shape matching is a long-standing problem in shape analysis. The goal is to find point-to-point correspondences between two different shapes.

We start by reviewing methods that directly predict dense correspondences via optimization on two shapes, then we review 3D shape descriptors, and correspondence methods in function space. Finally, we discuss approaches that solve the correspondence problem jointly on a collection of shapes, most related to our work.

Note that we simply give an overview of these different types of approaches, an exhaustive description of all methods is out of the scope of this thesis. For a more detailed survey the reader can refer to Tam et al. (2013); van Kaick et al. (2011).

Assumptions and shape representation. Throughout the chapter, we consider shapes that are 2D manifolds embedded in the 3D euclidean space. The length of the shortest curve between two points on such a 2D manifold is called the geodesic distance. We call "intrinsic" properties that are fully defined by the geodesic distance. We call "isometric" transformations that preserve the geodesic distances between any pair of points in a shape.

We seek to estimate correspondences between shape $\mathcal{X}$ and shape $\mathcal{Y}$. We assume that $\mathcal{X}$ and $\mathcal{Y}$ differ by a near-isometric transformation, except in the Iterative Closest Point method

where we assume they differ by a rigid transformation. The set of human shapes in different poses is of particular interest to this thesis. Note that two human shapes indeed differ by a near-isometric transformations, where the "near" comes from skin elasticity due to muscle contraction in a specific pose.

Depending of the approach, $\mathcal{X}$ and $\mathcal{Y}$ can be represented by point clouds, meshes or surfaces. Both point cloud and meshes are standard 3D data representations. Point clouds can always be sampled from meshes, and in turn meshes can be recovered from point clouds if they are dense enough and equipped with normals Kazhdan and Hoppe (2013). A common data source for meshes are Computer-Aided Design (CAD) models, though CAD models are often unclean and non-watertight.

Point-cloud based correspondence approaches can work directly on raw 3D scans of a physical object but cannot leverage shape properties like geodesic distances and curvature. These properties are useful to develop shape matching algorithms because they are invariant to isometric transformations.

Reasoning directly on surfaces is often an important step to develop a mesh-based correspondence approach. Geodesic distances, Gaussian curvature and the Laplace-Beltrami operator are typically first defined on a differentiable surface. The surface being discretized in a mesh, the operator on the mesh is built to be a good approximation of the operator on the differentiable surface. In the rest of the chapter, it will always be explicitly stated whether $\mathcal{X}$ and $\mathcal{Y}$ are represented by point clouds, meshes or surfaces.

2.2.1 Direct optimization

In this section, we discuss classical methods that compute correspondences by direct optimization on two shapes. We first review Iterative Closest Point (ICP), then minimum distorsion metric approaches. ICP operates under the assumption that the two shapes differ by a rigid deformation while minimum distorsion metric tackle the more general problem of near-isometric deformations.

2.2.1.1 Iterative Closest Point (ICP).

Proposed in the early nineties Besl and McKay (1992); Besl and McKay (1992); Chen and Medioni (1992), ICP is the base algorithm for rigid alignment of two point clouds. The spirit of this method is to iteratively find a rigid transformation *i.e.* a rotation R and a translation T that deforms the source point cloud $\mathcal{X}$ into the target points $\mathcal{Y}$. The iterations stop when a stopping criterion is met, typically when the reconstruction error no longer diminishes. The pseudo-code

for the algorithm is described in algorithm 1.

Algorithm 1: Iterative Closest Point

Data: Two **point clouds** $\mathcal{X}$ and $\mathcal{Y}$

while *Stopping Criterion not met* **do**

  (1) Match the point in $\mathcal{X}$ to the points in $\mathcal{Y}$ through nearest neighbors;

  (2) Find the best rotation R and translation T to apply on $\mathcal{X}$ that minimizes the pairwise distance between the matches with a least square minimization;

  (3) Apply R and T on $\mathcal{X}$;

end

Result: The set of matches between $\mathcal{X}$ and $\mathcal{Y}$, through nearest neighbors.

ICP works on raw point clouds and can generalize to n-dimensional point clouds. However, it only works if the two input shapes are already roughly aligned. Otherwise, it converges to a poor local minimum. Though the original ICP can only work if the two shapes differ by a rigid transformation, it was later extended by Amberg et al. (2007) to non-rigid alignment. The extension is also iterative in the sense that it alternates between rigid ICP steps and non-rigid parametric transformations. In chapter 5, we propose a method to train a deep neural network to deform any shape into any other shape from the same category. The neural network training is also an iterative process and uses a nearest-neighbor-based energy function, and can thus be interpreted as a neural ICP.

2.2.1.2 Minimum Distorsion Metric

ICP assumes that two shapes differ by a rigid transformations. The space of rigid transformations can also be defined as any transformation that preserves the euclidean distance between pairs of points. Another space that one can consider is the one that preserve geodesic distances on the shape *i.e.* isometric transformations. Consider for instance a sheet of paper that someone is bending: the euclidean distance between the opposite corners vary in 3D but their geodesic distance remain unchanged.

In the following, we briefly present three important strategies that have been developed to predict correspondence maps minimizing the deviation from isometry. Elad and Kimmel (2003) propose to lift each shape into a high dimensional space where euclidean distances between pairs of high-dimensional points is equal to the geodesic distance between these points, then apply rigid alignment methods in the high-dimensional space. Mémoli and Sapiro (2005) generalize this intuition and recast the objective as minimizing the Gromov-Hausdorff Distance.

Finally, Bronstein et al. (2006b) generalize previous approaches in a geodesic distorsion minimization framework. We detail those three approaches in the following paragraphs.

Multi Dimensional Scaling (MDS) To minimize the deviation of the correspondence map from isometry, Elad and Kimmel (2003) propose to embbed each shape independently in $\mathbb{R}^n$ with a transformation ϕ. ϕ is optimized such that Euclidean distances in the embedding space approximates geodesic distances on the shapes. Achieving this simplifies the problem: instead of searching for the best isometry in the 3D space, we now can look for the best rigid transformation in a high-dimensional space, with ICP for instance. There are several strategies to optimize ϕ. In the original MDS paper, Elad and Kimmel (2003) propose to minimize the *stress function*:

$$\text{stress}_p(\phi) = \left(\iint_{x_1,x_2 \in \mathcal{X}} \left(d_{\mathcal{X}}(x_1,x_2) - ||\phi(x_1),\phi(x_2)||_{L_2} \right)^p \mathrm{d}a_{x_1} \mathrm{d}a_{x_2} \right)^{\frac{1}{p}}, \tag{2.1}$$

where ϕ is a function from $\mathbb{R}^3$ to $\mathbb{R}^n$, x_1 and x_2 are a pair of points on shape $\mathcal{X}$, $d_{\mathcal{X}}(x_1,x_2)$ is the geodesic distance between them, $\mathrm{d}a_{x_1}$ and $\mathrm{d}a_{x_2}$ are infinitesimal surface elements on shape $\mathcal{X}$, p is an arbitrary integer (possibly taken as infinity which replaces the integrals by a maximum).

To find a minimizer of the stress function, Elad and Kimmel (2003) propose a least-squares method, and Ovsjanikov et al. (2008) use a spectral analysis of the Laplace-Beltrami operator. Elad and Kimmel (2003) suggest to choose the dimension n of the embedding space as the smallest integer such that the the optimum of the stress function is below a certain threshold. Figure 2.1 presents examples of optimized transformation ϕ on several shapes.

The Gromov-Hausdorff Distance As already mentionned, aligning shapes with MDS is a 2-step process: first find a minimizer of the stress function for each shape $\mathcal{X}$ and $\mathcal{Y}$, then align the embeddings with a rigid transform algorithm (ICP). Instead of a two-step process, Mémoli and Sapiro (2005) propose to jointly predict transformations for each shape, $\phi_{\mathcal{X}}$ for shape $\mathcal{X}$ and $\phi_{\mathcal{Y}}$ for shape $\mathcal{Y}$, and minimize jointly over $\phi_{\mathcal{X}}$ and $\phi_{\mathcal{Y}}$ the Hausdorff-distance $d_{\mathrm{H}}(\mathcal{X},\mathcal{Y})$ of their associated embeddings in $\mathbb{R}^n$. The optimum defines a distance between shape $\mathcal{X}$ and $\mathcal{Y}$, called the *Gromov-Hausdorff distance* $d_{\mathrm{GH}}(\mathcal{X},\mathcal{Y})$.

$$d_{\mathrm{GH}}(\mathcal{X},\mathcal{Y}) \equiv \inf_{\phi_{\mathcal{X}},\phi_{\mathcal{Y}}} d_{\mathrm{H}}(\phi_{\mathcal{X}}(\mathcal{X}),\phi_{\mathcal{Y}}(\mathcal{Y})) \tag{2.2}$$

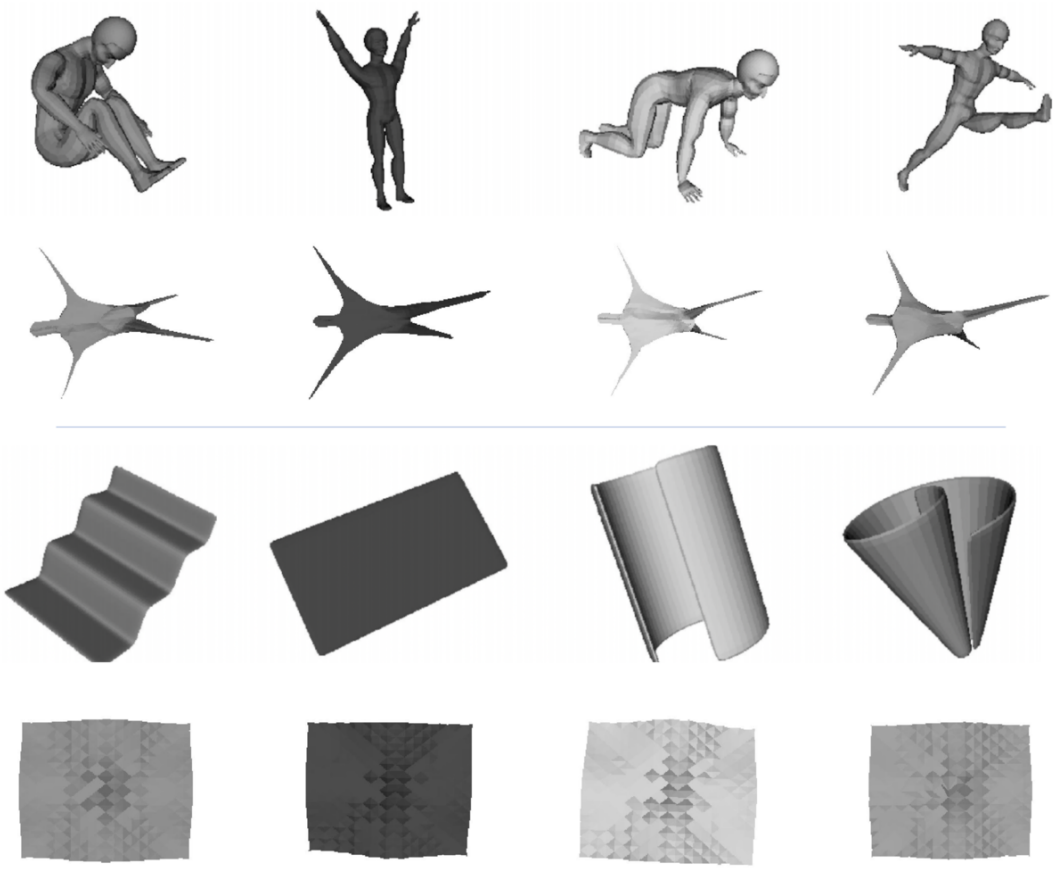


Figure 2.1 Figure from Elad and Kimmel (2003). Multi-dimensional scaling visualizations on bending versions of two shapes. The first raw shows four versions of isometric input shapes and the second raw is their MDS-mapping, where the dimensionality of the embedding space has been set to 3 for visualization purposes. Note the isometric shapes are mapped to similar embeddings.

Where d_{GH} is the Gromov-Hausdorff distance, $\phi_{\mathcal{X}}$ and $\phi_{\mathcal{Y}}$ are isometries from $\mathbb{R}^3$ equipped with geodesic distances to $\mathbb{R}^n$ equipped with euclidean distances, and d_{H} is the Hausdorff distance: the maximum distance of any point on surface $\mathcal{X}$ to any point on surface $\mathcal{Y}$.

The Gromov-Hausdorff distance can be interpreted as the Hausdorff distance between $\mathcal{X}$ and $\mathcal{Y}$ up to the isometries $\phi_{\mathcal{X}}$ and $\phi_{\mathcal{Y}}$.

Generalized MDS (GMDS) A problem of these embedding-based techniques is that they measure deviations from isometry only approximately in the embedding space. To avoid using explicitly the intermediate embedding space, the Gromov-Hausdorff distance can be expressed for bounded shapes with two transformations $\phi_{\mathcal{X}\to\mathcal{Y}}$ and $\phi_{\mathcal{Y}\to\mathcal{X}}$ respectively from $\mathcal{X}$ to $\mathcal{Y}$ and

$\mathcal{Y}$ to $\mathcal{X}$ as:

$$d_{\mathrm{GH}}(\mathcal{X},\mathcal{Y}) = \frac{1}{2} \inf_{\substack{\phi_{\mathcal{X}\to\mathcal{Y}} \\ \phi_{\mathcal{Y}\to\mathcal{X}}}} \max\{\mathrm{stress}_\infty \phi_{\mathcal{Y}\to\mathcal{X}}, \mathrm{stress}_\infty \phi_{\mathcal{X}\to\mathcal{Y}}, \mathrm{dis}(\phi_{\mathcal{X}\to\mathcal{Y}}, \phi_{\mathcal{Y}\to\mathcal{X}})\} \tag{2.3}$$

with

$$\mathrm{dis}(\phi_{\mathcal{X}\to\mathcal{Y}}, \phi_{\mathcal{Y}\to\mathcal{X}}) \equiv \sup_{x\in\mathcal{X}, y\in\mathcal{Y}} |d_{\mathcal{X}}(x, \phi_{\mathcal{Y}\to\mathcal{X}}(y)) - d_{\mathcal{Y}}(y, \phi_{\mathcal{X}\to\mathcal{Y}}(x))| \tag{2.4}$$

Where x and y are points on $\mathcal{X}$ an $\mathcal{Y}$ respectively, $d_{\mathcal{X}}$ (resp. $d_{\mathcal{Y}}$) is the geodesic distance on $\mathcal{X}$ (resp. $\mathcal{Y}$). $\mathrm{stress}_\infty \phi_{\mathcal{Y}\to\mathcal{X}}$ and $\mathrm{stress}_\infty \phi_{\mathcal{X}\to\mathcal{Y}}$ encourage isometric mappings while $\mathrm{dis}(\phi_{\mathcal{X}\to\mathcal{Y}}, \phi_{\mathcal{Y}\to\mathcal{X}})$ encourages $\phi_{\mathcal{X}\to\mathcal{Y}}$ and $\phi_{\mathcal{Y}\to\mathcal{X}}$ to be inverse of each other. Though this formulation avoids using explicitly an embedding space, Bronstein et al. (2006b) argues that it is is inappropriate to match partial shapes and restricts the objective to:

$$d_{\mathrm{GMDS}}(\mathcal{X},\mathcal{Y}) = \frac{1}{2} \inf_{\phi_{\mathcal{X}\to\mathcal{Y}}} \mathrm{stress}_\infty \phi_{\mathcal{X}\to\mathcal{Y}} \tag{2.5}$$

This simplification of the Gromov-Hausdorff distance leads to the generalized multidimensional scaling objective (GMDS) - also called generalized stress function as:

$$\phi^*_{\mathcal{X}\to\mathcal{Y}} = \arg\min_{\phi_{\mathcal{X}\to\mathcal{Y}}} \left(\iint_{x_1,x_2\in\mathcal{X}} \left(d_{\mathcal{X}}\left(x_1,x_2\right) - d_{\mathcal{Y}}\left(\phi_{\mathcal{X}\to\mathcal{Y}}\left(x_1\right), \phi_{\mathcal{X}\to\mathcal{Y}}\left(x_2\right)\right)\right)^p \mathrm{d}a_{x_1} \mathrm{d}a_{x_2} \right)^{\frac{1}{p}} \tag{2.6}$$

When $\mathcal{X}$ and $\mathcal{Y}$ are representation by meshes with n points, solving the GMDS is a quadratic assignment problem (QAP), where the minimum is sought over the space of $n \times n$ permutation matrices. Note that while the GMDS is hard to solve, it is simply based on a measure of distorsion, making it a natural candidate for learned methods without supervision as it does not require any annotated correspondences. In Chapter 4, we use a similar loss to regularize parametric deformations to learn unsupervised correspondences.

Solving the GMDS in practice Several approaches have succeeded in reducing the complexity of this QAP problem.

- **Convex relaxations.** Chen and Koltun (2015) propose a convex relaxations of the permutation matrix by considering soft assignments instead of hard assignments. They solve the dual of the relaxed problem with a Linear Program (LP) solver. Rodolà et al. (2012) propose another relaxation based on game-theory of a variant of the GMDS called

the Lipschitz formulation obtained by replacing the geodesic distances $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$ by log distances. Their relaxation outputs a sparse set of reliable correspondences. Like a number of other methods optimizing for soft correspondence, this does not usually scale to full-resolution of discretized models, but can be particularly useful to initialize dense methods. On the contrary, our method in chapter 4 predicts dense correspondences.

- **Hierarchical matching.** Sahillioglu and Yemez (2011) propose to reduce the search space of $n \times n$ permutation matrices by exploiting the fact that the optimal mapping is found among functions that maps nearby vertices on the source shape to nearby vertices on the target. Hence the matching can be performed in a coarse-to-fine fashion. After solving the full problem on few salient points, they use coarse solutions to guide the optimisation for gradually larger problem until dense matching. This idea is also present in D.Raviv et al. (2013) and Bronstein et al. (2006a) to remedy the convergence problem involved in the optimization of the stress function.

- **Conformal Maps.** Kim et al. (2011); Lipman and Funkhouser (2009) also propose to reduce the search space to the space of conformal mappings. Conformal maps are functions that locally preserve angles and contain the space of isometric transform as a sub-space. Polynomial-time search algorithms for the GMDS are available in the space of conformal maps. A practical example of this idea is the Mobius Voting strategy from Lipman and Funkhouser (2009). Their appraoch exploits the fact that the group of conformal maps between two sphere, called the Mobius group, is parametrized by 6 parameters. Estimating the these 6 parameters can be done with a Hough voting strategy. To estimate a conformal map from $\mathcal{X}$ to $\mathcal{Y}$, they map both shapes to a sphere with conformal maps then solve the problem on the sphere. This is illustrated in Figure 2.2. Kim et al. (2011) propose to increase the expressivity of the conformal map model by blending several conformal maps together.

Though well-posed theoretically, minimum distorsion metric approaches are hard to optimise and often lead to a poor local minima Bronstein et al. (2006a). Inspired by the success of 2D descriptors, a complementary approach to approaches based on minimum distorsion metric has been to match 3D shape descriptors.

2.2.2 Local correspondence by shape descriptors.

Finding good 3D descriptors has been a very active area of research. Local shape descriptors are discriminative embeddings of the neighborhood (potentially the full shape) of a point on a shape. Descriptors are often used to obtain a sparse set of reliable correspondences to initialise

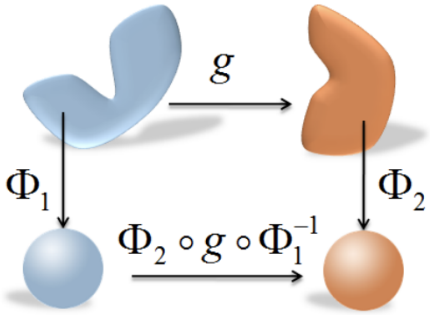


Figure 2.2 Figure from Lipman and Funkhouser (2009). After mapping both genus-0 shapes $\mathcal{X}$ and $\mathcal{X}$ to a sphere with conformal maps $\Phi_{\mathcal{X}}$ and $\Phi_{\mathcal{X}}$, the problem of finding the conformal mapping g between $\mathcal{X}$ and $\mathcal{X}$ is translated to finding the 6 parameters of the conformal $\Phi_{\mathcal{Y}} \circ g \circ \Phi_{\mathcal{X}}^{-1}$ between the two spheres.

dense methods. Thus, ideal shape descriptors should be: discriminative, concise, and cheap to compute, and robust (*i.e.* have some notion of invariance).

We start by explaining how to use descriptors to get a set of reliable correspondences via outlier rejection. We then discuss 3D descriptors starting with three classical descriptors: Spin Images Johnson (1997), 3D Shape Context Belongie and Malik (2000); Kokkinos et al. (2012), and Shape HOG Zaharescu et al. (2009) descriptors. Then we review three spectral descriptors: the Global Point Signature Rustamov (2007), the Heat Kernel Signature Sun et al. (2009a) and the Wave Kernel Signature Aubry et al. (2011) and discuss an early approach that learn optimal spectral descriptors Litman and Bronstein (2013).

Our approach developed in Chapter 4 for shape matching do not rely on shape descriptors, but we compare against approaches using shape descriptors and outperform them.

2.2.2.1 Sparse set of reliable correspondences by outlier rejection

A possible strategy to extract a sparse set of reliable correspondences from shape descriptors is to use an outlier rejection method like RANSAC Fischler and Bolles (1981). Indeed, considering the set of nearest-neighbor pairs on the descriptors, the quality of a sub-set of correspondences can be measured via their induced distorsion on the whole set with the GMDS. One can thus sample subsets and select the subset that induces the lowest distorsion, which should not contain any outlier. Getting a good sparse set of correspondences is critical to initialize approaches solving the GMDS problem. Indeed, the GMDS problem is non-convex, so the optimization is sensitive to an initial guess Bronstein et al. (2006a).

2.2.2.2 Classical descriptors

We start by reviewing three classical shape descriptors: Spin Images Johnson (1997), 3D Shape Context Belongie and Malik (2000); Kokkinos et al. (2012), and Shape HOG Zaharescu et al. (2009) descriptors.

Spin image. This idea was introduced by Johnson (1997); Johnson and Hebert (1999) for point clouds with normals. For each point, a cylindrical coordinate system is defined with the described point at the center and its normal as the cylinder axis. A 2D histogram of point density is then computed in a predefined neighbourhood on the elevation and radius of each neighbor point.

Shape Context. The shape context descriptor was originally designed for images by Belongie et al. (2002) and was later extended to 3D meshes by Kokkinos et al. (2012); Körtgen et al. (2003). Given a mesh, a histogram is computed on the log-polar coordinates of neighboring points using geodesic distances. To solve the orientation ambiguity, the final descriptor is the modulus of the Fourier transform of the image histogram.

Shape HOG. Like Shape Context, shape HOGs Zaharescu et al. (2009) computes histograms in log-polar coordinates. However, it discriminates on texture information instead of density of points. To achieve this, it stores in each bin of the histograms the dominant gradient orientations of the projected texture. This descriptor assumes that textures are available while the other discussed descriptors do not.

2.2.2.3 Spectral descriptors

Spectral descriptors are based on the Laplace-Beltrami (LB) operator on a surface. The eigenvalues and eigenfunctions of the LB operator are invariant to isometric transformation of the surface. The eigenfunctions also form a basis of the space of functions on the surface. Any linear combination of those eigenfunctions is therefore intrinsic.

Global Point signature (GPS). The GPS Rustamov (2007) of a point is the concatenated values of the first eigenfunctions at that point scaled by the the square root of the norm of the eigenvalue. This descriptor is not very robust to slight modification of the shape, which can change the order of the eigenfunctions.

The Heat Kernel Signature (HKS). The HKS Sun et al. (2009b) is a standard spectral shape signature. The HKS uses low-pass filters. The idea is based on the physical link between the Laplace-Beltrami operator and the heat diffusion process. The HKS descriptor at point x has a physical interpretation : it measures the quantity of heat at position x and time t after heating exactly and only that point at time 0. The HKS was extended to model a volumetric heat diffusion process by D.Raviv et al. (2013) and a scale-invariant HKS was designed by Bronstein and Kokkinos (2010).

Wave Kernel Signature (WKS). Similar to HKS, the WKS Aubry et al. (2011) can be seen as a family of filters applied to the LB eigenfunctions, but while the HKS uses low pass filters, the WKS uses band filters and thus tends to be more discriminative than the HKS.

Optimal Spectral descriptors. To improve spectral shape descriptors, a natural idea is to replace hand-crafted linear combination of LB eigenfunctions with learned combinations. This idea has already proved successful in image matching: Brown et al. (2011); DeTone et al. (2017); Dusmanu et al. (2019); Luo et al. (2019); Revaud et al. (2019); Zagoruyko and Komodakis (2015) learn data-driven 2D descriptors and demonstrate their superiority over handcrafted 2D descriptors. Litman and Bronstein (2013) propose a parametric family of transfer functions, formed with the Laplace-Beltrami eigen-functions, which includes the HKS and the WKS as particular instanciations. They propose to learn an optimal spectral descriptor from that family using a joint optimization on a collection of human shapes. Their optimized descriptors shows noticeable improvement upon handcrafted descriptors. Further approaches leveraging shape collections to learn local shape descriptors, in particular using deep learning techniques, are discussed in Section 2.2.4.3.

2.2.3 Correspondence in function space.

Instead of mapping points, Ovsjanikov et al. (2012) propose to map functions between two surfaces $\mathcal{X}$ and $\mathcal{Y}$. We start by discussing how to build a functional representation of the correspondence problem, and present its advantages. Finally, we discuss deep approaches learning functional maps on shape collections.

2.2.3.1 From point correspondences to function correspondences.

Given point-to-point correspondences, one can get function-to-function correspondences. Indeed, given point-to-point correspondences defined by a bijection $T : \mathcal{X} \to \mathcal{Y}$, we can associate to any scalar function $f : \mathcal{X} \to \mathbb{R}$, the function g on $\mathcal{Y}$ defined as $g = f \circ T^{-1}$. This induces

correspondences on the space of all scalar functions $\mathcal{F}(\mathcal{X},\mathbb{R})$ abstracted by the functional $T_F : \mathcal{F}(\mathcal{X},\mathbb{R}) \to \mathcal{F}(\mathcal{Y},\mathbb{R})$.

To get the correspondences of a point given a functional representation, one can apply the functional on the Dirac function. However, while one can associate to each bijection T a functional T_F, there exist functionals that do not correspond to bijective transformations between the shapes. Thus, given an arbitrary functional, a Dirac can be associated to a distribution which is not a Dirac and further post-processing is needed to extract point-to-point correspondences.

The key advantage of a functional representation is that the functional T_F is a linear operator in the space of functions $\mathcal{F}(\mathcal{X},\mathbb{R})$. As a consequence, most natural constraints on a mapping, such as descriptor preservation, landmark correspondences, part preservation and distorsion minimization become linear in this formulation. On the contrary, incorporating these constraints in a direct point-to-point approach leads to difficult non-convex optimizations. Solving the mapping functions also enables function transfer in a collection of shapes like part segmentation without establishing point to-point correspondences.

2.2.3.2 The functional map as a matrix.

The space of functions $\mathcal{F}(\mathcal{X},\mathbb{R})$ and $\mathcal{F}(\mathcal{Y},\mathbb{R})$ can be decomposed in an orthonormal basis. A simple example of such a basis is the family of all Diracs. Since the operator T_F is linear, the problem of matching all functions can thus be reduced to matching the base functions. While the number of base functions can be infinite, this formulation allows flexibility in the choice of orthonormal basis. For surfaces, the Laplace-Beltrami eigenbase is a natural extension of the Fourier orthonormal base. The first eigenfunctions of this basis represents well low-frequencies. In fact, Aflalo et al. (2014) have shown that Laplacian eigenbases are optimal for representing smooth functions on a surface. This means that by restricting the linear decomposition of a function to the first eigenvalues, we apply a low-pass filter on the fonction that preserves the low frequency and mid frequency components and lose the high-frequencies. This approximation makes the problem of aligning the basis functions computationally tractable, and the functional map can be represented a matrix of correspondences between the base functions, as illustrated in Figure 2.3.

One of the main limitation of functional maps is that the functional representation, especially restricted the first functions of a basis, do not directly give point-to-point correspondences.

2.2.3.3 Deep functional maps.

Functional maps are a powerful generic tool working for a large variety of shapes. Many shape correspondence approaches build on functional maps, and a full review is outside the scope of

$$T_F(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 T_F(f_1) + \alpha_2 T_F(f_2)$$

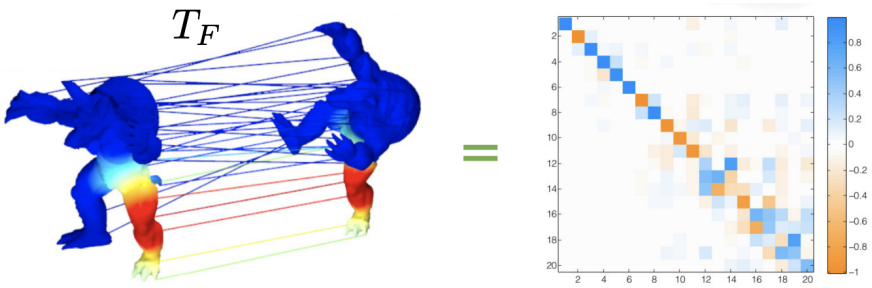


Figure 2.3 Figure from http://www.lix.polytechnique.fr/ maks/fmaps_SIG17_course/slides/lecture0.pdf. Point-to-point correspondences between shapes (surface or pointcloud) gives function-to-function correspondences. The induced functional map T_F mapping functions on a shape to functions on the other shape is a linear operator which can be represented as a matrix given a basis.

this work. However, we present quickly two approaches exploiting functional maps on shape collections that are the most related to our work. Note that section 2.2.4 will discuss more generally learning correspondences on shape collections.

Similarly to Litman and Bronstein (2013), Litany et al. (2017) learn spectral shape descriptors using the functional representations between all pairs in a shape collection. They propose to first refines an initial shape descriptor with a shared neural network across shapes. This neural network is a shared Multi Layered Perceptron (MLP) that takes as input the multi-dimensional descriptor of a point on a shape and applies a non-linear function on it. In their approach, this MLP is the only trained module and aim enhance the initial descriptor into a refined descriptor. After this step, the refined descriptor is projected on the basis of functions used by the matrix-based functional representation. This gives a vector in the functional representation of the refined descriptor applied on a point of a shape. Corresponding descriptors can finally be matched through linear matrix constraints. Their key insight is that the pipeline predicting the mapping between two shapes given shape descriptors and ground-truth point-to-point correspondences is end-to-end differentiable, so the shared MLP can be trained via backpropagation on a collection of 100 human shapes with correspondence annotations from the FAUST dataset Bogo et al. (2016). Halimi et al. (2019) extended this idea to an unsupervised setting by minimizing only for low-induced distorsion by the functional.

Both approaches share the same limitations as the original functional map approach: the resulting functional map only aligns basis functions and additional optimization is required to extract consistent point-to-point correspondences, or a smooth template deformation for surface

reconstruction Ovsjanikov et al. (2012). These method also relies on a hand-crafted point-wise descriptor as initialization Tombari et al. (2010) and use neural networks to improve the descriptor. In contrast, in Chapter 5, we introduce a method that does not rely on hand-crafted features (it only takes point coordinates as input) and directly outputs a template deformation.

2.2.4 Shape matching in collections.

Some common categories, such as humans, benefit from a profusion of existing data Bogo et al. (2014); Zuffi et al. (2017) and can leverage strong class-specific priors for shape matching through the use of an explicit template. We start this section by briefly describing how to construct and use a morphable template for the category of human scans. Another approach to leverage class-specific knowledge is to run the direct optimization approaches jointly on many pairs of shape from the same category. We discuss in particular an approach leveraging cycle-consistency on the collection to improve individual mappings by Nguyen et al. (2011). Finally we present learning approaches that cast shape matching as a classification problem called the "labeling problem".

2.2.4.1 Template-based shape matching for humans

Correspondences of humans shapes is of particular interest to this thesis since the algorithm we develop for shape matching in chapter 4 is compared against other approaches on a benchmark of human correspondences. For humans, shape matching approaches using a morphable model of the human body outperform direct optimization approaches. We first discuss how to construct such a model, then explain how to fit the model to new shapes to achieve correspondences.

Creating a morphable template. Creating a morphable template is a difficult task which took more than a decade of research to reach the current level of maturity Allen et al. (2002, 2003, 2006); Loper et al. (2015); Zuffi and Black. (2015). The Skinned Multi-Person Linear (SMPL) model by Loper et al. (2015) is currently a standard human model. SMPL is a parametric mesh model with two sets of control parameters, θ and β. θ controls the pose of the human while β controls the shape appearance of the model. The full spectrum of pose and shapes can be explored by varying θ and β. Figure 2.4 shows 25 samples. We use the SMPL model in Chapter 4 to generate training data for shape matching with dense correspondence labels.

SMPL is based on several ideas. Starting from a base mesh in a resting pose, each vertex on the mesh is linearly linked to a skeletal structure. The pose parameters θ control the axis-angle rotation parameter of each joint with respect to its ancestor in the skeletal structure. This is

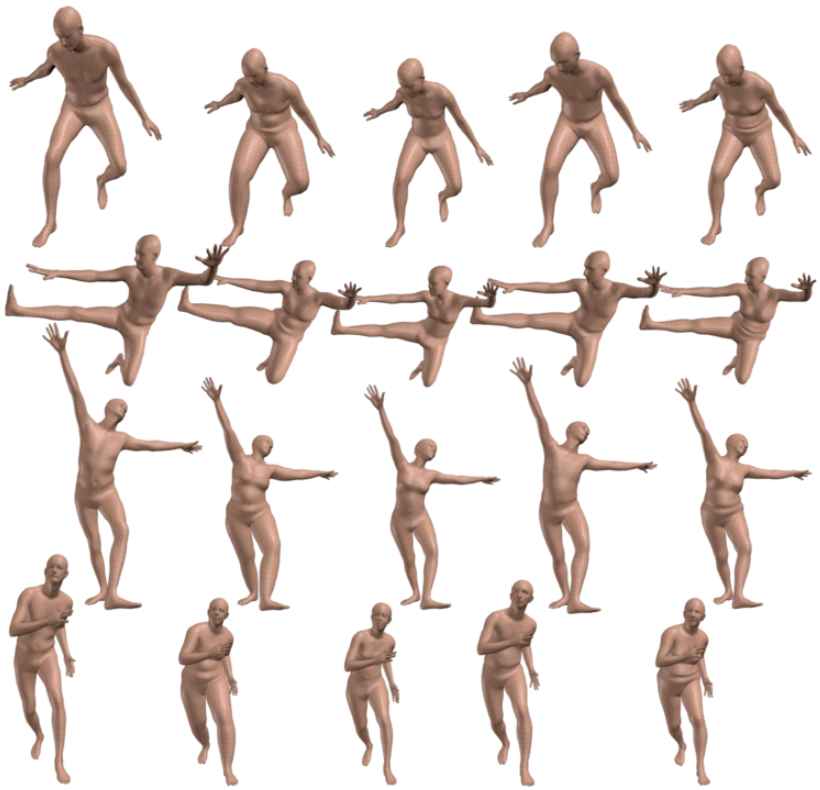


Figure 2.4 Figure from Loper et al. (2015). Samples from the SMPL model. Decomposition of SMPL parameters into pose and shape: Shape parameters β vary across different subjects from left to right, while pose parameters θ vary from top to bottom for each subject.

known as Blend Skinning Wang and Phillips (2002). The human body has local variations depending on pose and shape. To account for these variation, the base shape is updated depending on the pose parameters θ, and blended with a linear combinations of diverse shapes controlled by parameter β. The general idea is to slightly modify the base shape prior to blend skinning in order to minimize the artifacts Lewis et al. (2000).

Recently, Pavlakos et al. (2019) has extended the SMPL model with fully articulated hands and an expressive face. Zuffi et al. (2018, 2017) has extended the SMPL model to animals including lions, cats, tigers, dogs, horses, cows, foxes, deers, zebras, and hippos.

Using class-specific models for matching. Fitting the SMPL model to a given shape is not convex with regard to parameters (β, θ) which makes it hard in practice. To reach a good local minimum, the fitting objective function is typically regularized. Bogo et al. (2016) propose two

regularization terms: an interpenetration penalization of the body parts, and a prior for realistic poses and shape learned with a Variational AutoEncoder (VAE).

In chapter 4, instead of learning SMPL parameters (β, θ) for human reconstruction, we let a deep neural network learn how to deform a base human mesh.

2.2.4.2 Joint optimization with cycle-consistency

Shapes categories exhibiting a large degree a topological variation, like chairs, do not have a clear template. Class-specific context can still be exploited by jointly optimizing shape matching on the entire shape collection while encouraging cycle-consistency of the correspondence maps Huang and Guibas (2013); Huang et al. (2012); Kim et al. (2012); Nguyen et al. (2011); Rustamov et al. (2013). In particular, given a collection of shapes and estimated maps between them Nguyen et al. (2011) construct of graph where each node is a shape. Edges between shapes are scored using the average deviation of 3-cycles involving the edge. If the map associated to that edge is poor, then all 3-cycles involving it will have poor cycle-consistency and thus the average will have a high error score. Conversely, if the map is accurate, and the ratio of accurate mappings is sufficiently high in the graph, then the average error score will be low. In the end, maps are improved by replacing the original maps with composed maps along the shortest paths in this graph. Composing maps is easy to do in the functional map framework since it is a matrix multiplication of the two functionals.

Joint optimization techniques are very powerful, but involve optimizing for many degrees of freedom with complex non-convex objective functions, and takes minutes or hours. To make matters worse, joint analysis usually scales in a super-linear manner with the number of shapes, and if a new shape is added to a collection, the entire optimization needs to be repeated.

Recently, learning-based correspondence techniques were used to address these limitations. They are fast, typically only requiring a forward pass through a neural network, and they enable joint analysis of a collection of shapes, since multiple shapes are typically used during training.

2.2.4.3 Learning correspondences through the labeling problem.

Rodola et al. (2014) introduce the labeling problem in one of the first learning-based method for shape matching. The labeling problem views shape matching as a classification task where the goal is to predict the index of each point on a common template. Matching two shapes can thus be done by labeling both shapes. The labeling problem is learned on a training set of annotated shapes with a cross-entropy loss comparing predicted labels and ground truth labels, which is standard for classification. Rodola et al. (2014) learn the labeling problem with a "shallow" random forest applied applied on WKS descriptors. In addition to the classification loss, the

distorsion induced by the predicted labels is minimized by a regularization loss formulated in the functional map framework.

Recent trends to learn shape matching solve the labeling problem with deep neural networks and can be categorized between intrinsic and extrinsic methods. Intrinsic approaches operate on surfaces and are invariant to isometric deformations of the surface. Note that the optimal spectral descriptors discussed in 2.2.2 and deep functional maps discussed in 2.2.3 are intrinsic methods. We first focus on intrinsic approaches based on non euclidean CNNs. We then discuss another line of work, called here extrinsic deep matching, that operates on the 3D space and are not invariant to isometric transformation. In the literature, extrinsic deep approaches often do not solve the labeling problem but solve the simpler classification problem of part labels for each points. In Chapter 4, we introduce a new extrinsic deep approach.

Deep Intrinsic Approaches based on Non-Euclidean CNNs. An important effort has been made to generalize CNN on 2D regular grids to non-Euclidean domains (surfaces). Convolutions on meshes are only a particular case of general graph neural networks, which has received a lot of attention Bronstein et al. (2017); Kipf and Welling (2017); Simonovsky and Komodakis (2017). We refer readers to Wu et al. (2019) for a recent survey on the subject. Like convolutions in the euclidean 2D plane, convolutions on arbitrary surfaces can either be done in the spatial domain by sliding a kernel on the shape, or in the spectral domain by applying a filter on the Laplace-Beltrami operator eigenfunctions. We start by discussing spatial approaches then spectral approaches.

Spatial CNN. Masci et al. (2015) were the first to introduce a geodesic CNN model operating on meshes, which performs non-euclidean convolutions by sliding a window over the manifold. The key consideration is how to extract a patch around a point. In Masci et al. (2015), local geodesic coordinates in a bounded radius are used in place of image 'patches'. This can be seen as an adaptation of the shape context descriptor to neural networks. Like shape context Belongie et al. (2002), the polar coordinates are defined up to arbitrary rotation $\theta \in [0, 2\pi[$ due to the ambiguity in the selection of the origin of the angular coordinate. The ambiguity is resolved by taking the maximum over all possible rotations of the extracted patch R_θ. Alternatively, Boscaini et al. (2016a) (ACNN) use anisotropic LB operators Andreux et al. (2014) to extract intrinsic patches on manifolds. Finally, Monti et al. (2017) present MoNet, a generalization of previous hand-crafted patch extractor and learn an optimized patch extractor. This extends the framework to arbitrary graphs. Figure 2.5 illustrates how these spatial non-euclidean CNNs extract a neighboorhood around each point. Figure 2.6

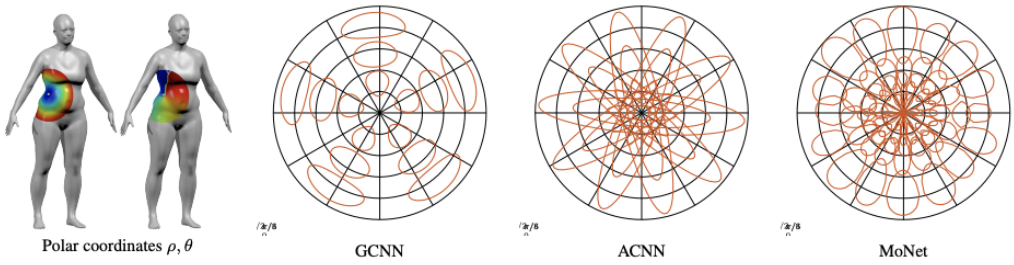


Figure 2.5 Figure from Monti et al. (2017). *Left*: intrinsic local polar coordinates ρ, θ on manifold around a point marked in white. *Right*: patch kernels used in different generalizations of convolution on the manifold (hand-crafted in GCNN and ACNN and learned in MoNet). Red curves represent the 0.5 level set.

compares non-euclidean CNN methods with Blended Intrinsic Maps Kim et al. (2011) for human correspondences, showing the superiority of learned approaches.

Spectral CNN. Bruna et al. (2014) defined a generalization of convolution in the spectral domain. In the Euclidean case, the Convolution Theorem states that the convolution operator is diagonalized in the Fourier basis. Similarly on a surface one can define a non-shift-invariant convolution by multiplying the spectral decomposition of a function in the Laplace-Beltrami basis. Spectral kernels lack spatial localisation and are harder to interpret than spatial kernels. The main limitation of spectral approaches is that the kernel depends on the Laplace-Beltrami basis and thus generalize poorly to different shapes. To mitigate this drawback and generalize better, Yi et al. (2016b) propose to transport back and forth the basis to a canonical base of functions shared across the shape collection and apply the kernels in the canonical base.

Deep Extrinsic Approaches for shape segmentation. Extrinsic 3D shape encoders have tremendously progressed in the past 5 years. These neural networks are not applied to the true labeling problem problem but rather to the task of labelling shape parts, and only recently to fine-grained segmentation Mo et al. (2019b). A full survey is out of the scope of this thesis, but we highlight the milestones in the following. Extrinsic shape encoders can be divided depending on how they represent 3D shapes. We focus on neural network that analyse volumetric representations (voxels) and point clouds.

Volumetric methods. Voxels are the natural extension of pixels in 3D. A general advantage of this choice of representation is that most pixel-based operators are easy to generilize to voxels. Choy et al. (2016) propose to use 3D convolutions on the voxel-grid to perform inference on

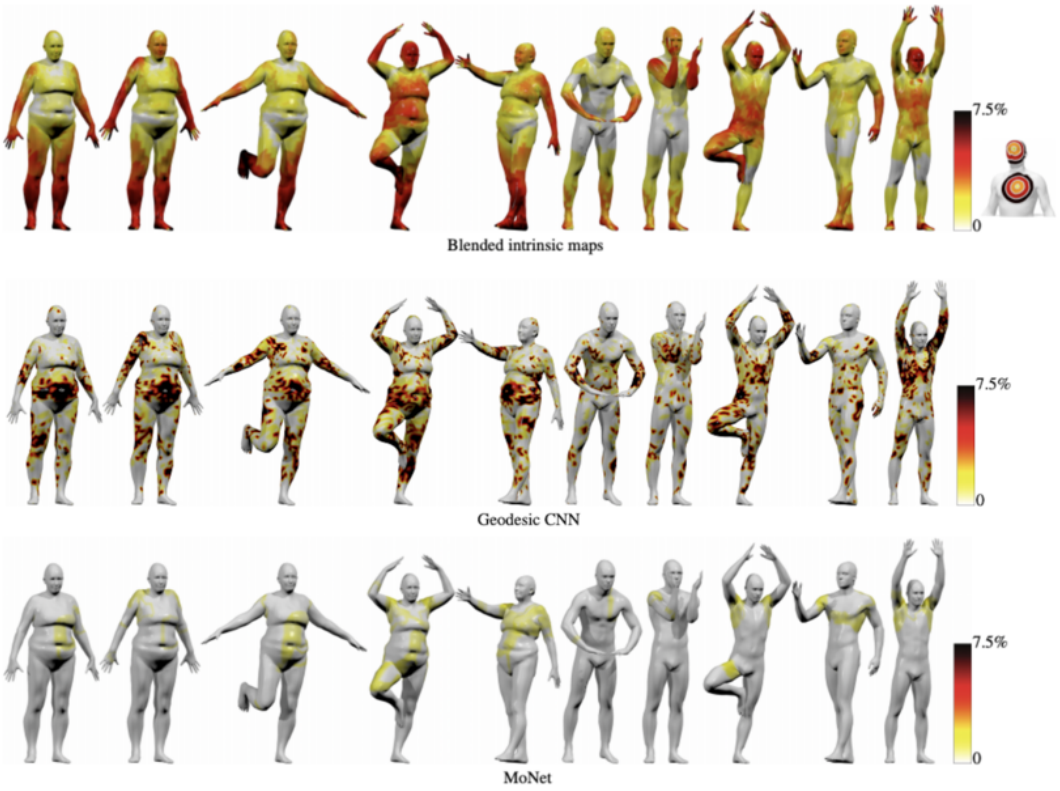


Figure 2.6 Figure from Monti et al. (2017). Pointwise error (geodesic distance from groundtruth) of different correspondence methods on the FAUST humans dataset, demonstrating the strenghts of deep learned approaches over direct optimization methods.

volumes. The main limitation of this approach is that 3D convolution are memory-hungry and this limits the analysis to low spatial resolution. Riegler et al. (2017) propose an octree-based optimization of the 3D convolutions operator based on the observation that the output of a 3D convolutional kernel is the same inside an octree region and only varies at the border. They demonstrates results on 3d facade part segmentation, however the facades are initially aligned which is a favorable setting for extrinsic approaches.

PointCloud methods. The main challenge to process point clouds with deep networks is to design architectures that are invariant to any permutation of the points. It has been an open challenge until the pioneering work of PointNet Qi et al. (2017a). To achieve this invariance, the main idea of PointNet is to stack: a high dimensional embedding by feeding each point to a shared MLP, and a symmetric function that aggregates information across points, typically a max operation. This simple yet powerful idea had an incredible impact in the community and spanned many follow-ups.

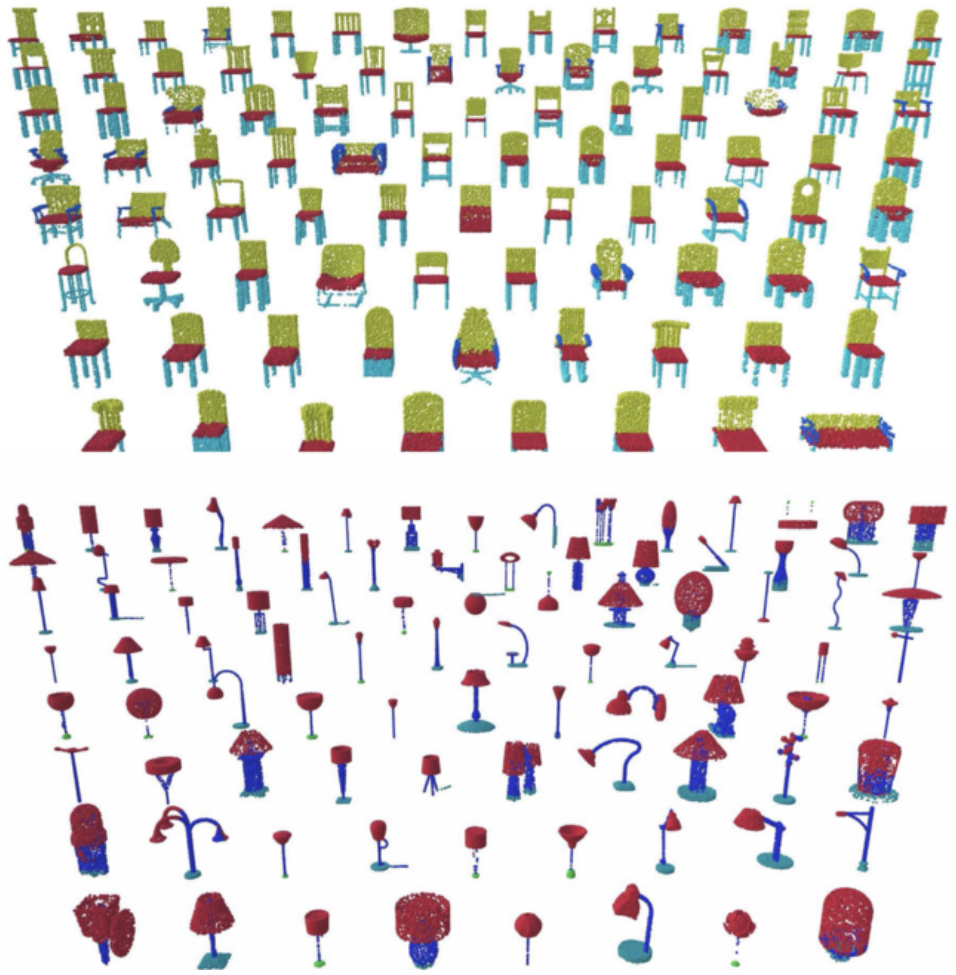


Figure 2.7 Figure from Wang et al. (2018b). Dynamic Graph CNN part correspondence results for chairs and lamps

While aggregating global information with a max function is appealing because of simplicity, it is also one the major limitation of PointNet. Inspired by the successes of hierarchical filters in images, the same group of authors Qi et al. (2017b) design a hierachical PointNet called PointNet++. The idea is to stack shared PointNet networks applied on the neighbourghood of each point. Though simple, it is hard to put in practice and requires careful tuning of many parameters. Like for non-euclidean CNNs, how to define a good neighbourghood is a key question. PointNet++ uses a ball of constant radius while Dynamic Graph CNN (DGCNN) Wang et al. (2018b) uses the k-nearest neighbors. To get all the points inside the ball radius or the k-nearest neighbors, PointNet++ uses the Euclidean metric on the spacial position of the points while Wang et al. (2018b) uses the Euclidean metric on the current embeddings. PointNet++ and DGCNN demonstrate state-of-the-art part segmentation of difficult categories from the ShapeNet dataset Chang et al. (2015), exhibiting heavy topological variations (see

Figure 2.7). Compared to the initial PointNet, they are also significantly heavier in memory which explains why PointNet remains a neural architecture of choice in many pipelines: a simplified PointNet architecture is used in all the chapters of this thesis.

Despite tremendous progresses, extrinsic deep methods typically only achieve shape segmentation. In contrast, in chapter 4 and 5, we present the first extrinsic deep methods to perform dense matching.

2.3 Single-view reconstruction

Taking a single image and estimating the physical world which produced that image is a fundamental problem in computer vision. It dates back at least to Lawrence G. Robert who phrase his goal almost 60 years ago as: "to make it possible for a computer to reconstruct and display a three-dimensional array of solid objects from a single photograph". This *inverse graphics* problem is underconstrained. Indeed, many different combinations of textures and lightning conditions can reproduce a specific image.

Adelson and Pentland (1996) have a beautiful metaphor to illustrate that many physical interpretations of a single image are possible. Looking at the image in Figure 2.8, different artists have different ways of explaining the scene depending on their own biases. A painter would see a canvas painted with the image, a sculptor would see it as an arrangement of bent shapes and a gaffer would see texture-less planar surface lit by a arrangement of various lights. In between those three extreme explanations exists a range of possible explanations, in which lies the most likely solution seen in Figure 2.8.b: A twice bent planar surface with a stroke of paint.

This problem is known to be underconstrained at least since the 11th century. Studying how the human visual system operates, the scientist Alhazen states in his "Book of Optics" that: "Nothing of what is visible, apart from light and color, can be perceived by pure sensation, but only by discernment, inference, and recognition, in addition to sensation". This hints at a statistical formulation of the problem. Among all possible physical explanation of an image, the goal is to find the one that maximizes a particular prior on 3D shapes, in other words, the most likely explanation based on past observations.

First, we discuss approaches estimating a depth map from a single-image. Classical methods tackle this problem through the "intrinsic image problem" which aims at explaining a single image by a depth map image, a reflectance image, and an illumination model Horn (1974). We also discuss learning approaches to predict depth maps. Depth maps however only describe the visible part of a photographed object. Second, we discuss approaches that rely on one or several 3D templates to estimate the full shape of a object from a single image. Last, we discuss

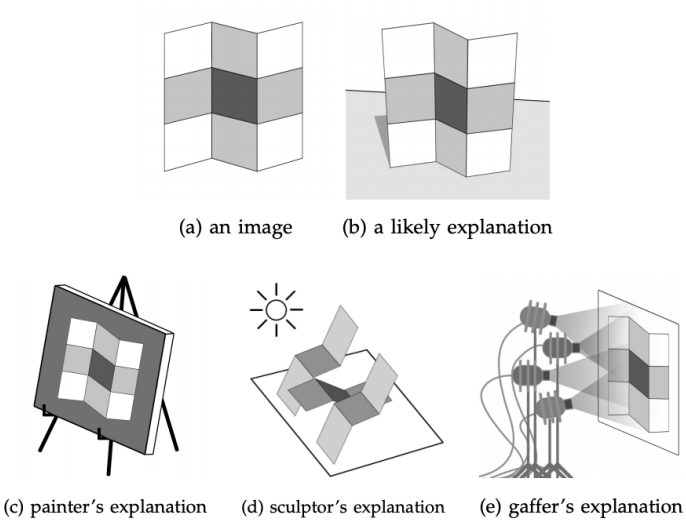


Figure 2.8 Figure from Barron and Malik (2015). A visualization of Adelson and Pentland's "workshop" metaphor Adelson and Pentland (1996). 1(b) is the most likely interpretation of the image in 1(a), but it could be a painting, a sculpture, or an arrangement of lights

recent deep approaches that learn implicit shape priors from data collections instead of using templates and reconstruct 3D shapes using volumetric functions, point-clouds or meshes.

Note that AtlasNet, our approach to SVR developed in Chapter 3, also learns implicit priors with deep networks and reconstruct shapes by predicting surface deformations.

2.3.1 Depth from a single image

We start by discussing the classical "intrinsic image" formulation of the SVR problem, then discuss approaches that learns to predict a depth map from an image.

2.3.1.1 Intrinsic image decomposition

In 1978, Barrow and Tenebaum defined the problem of "intrinsic images": recovering shape, reflectance, and illumination from a single image Barrow et al. (1978). Solving the intrinsic image problem enables image editing application such as shadow removal Kwatra et al. (2012), image colorization Liu et al. (2008), image re-texturing Carroll et al. (2011), and scene relighting Duchêne et al. (2015). A complete presentation of these applications is outside the scope of this thesis but some applications from Barron and Malik (2015) are illustrated in figure 2.9 and the reader can refer to Bonneel et al. (2017) for a good survey.

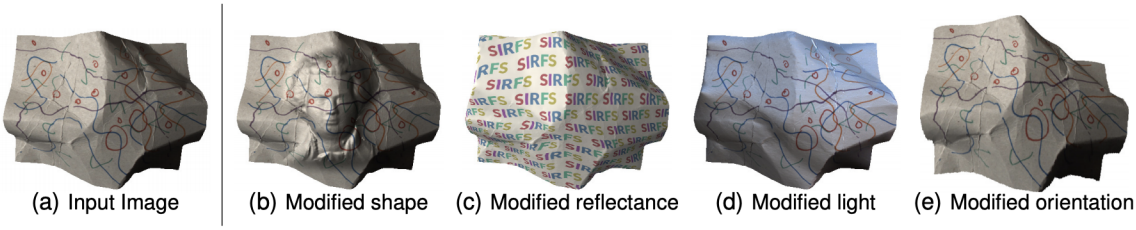


Figure 2.9 Figure from Barron and Malik (2015). Graphics applications of intrinsic image decomposition. Given only a single image, the algorithm estimates an object's shape, reflectance, or illumination. Any of those three scene properties can then be modified.

Two specific simplified approaches of this complex inverse problem are particularly relevant to us in the context of SVR:

1. Overloading terminology, **"intrinsic images"** has evolved in the problem of separating an image I into a shading image S and a reflectance image R linked by the following equation:

$$I = R \cdot S \tag{2.7}$$

   Such a separation in reflectance and shading assumes a Lambertian model of light diffusion, do not account for color bleeding nor specular surfaces. In this problem, the number of unknowns is twice the number of equations.

2. **Shape from Shading**: recovering the shape of an object given a single image, assuming illumination and reflectance are known. The original formulation of the problem is from Horn (1975). A more detailed survey of classic shape from shading methods can be found in Horn and Brooks (1989); Ruo Zhang et al. (1999).

Both problems need to be solved to go from a single image to a 3D shape but both problems are ill-posed. Figure 2.10 illustrates two well-known sources of ambiguity for Shape-from-shading: the bas-relief ambiguity, and the convex-concave ambiguity. As noted by Koenderink et al. (1996), both computational algorithms and the visual system face these ambiguities and have two means to deal with them: use some prior knowledge about the world Ikeuchi and Horn (1981), or use more than a single image. Approaches using more than a single-image to reconstruct shape, like an image from another viewpoint Hartley and Zisserman (2004); Triggs et al. (2000), or with different lightning conditions Basri and Jacobs (2001); Woodham (1992) are beyond the scope of this thesis. We now discuss classical assumptions used in optimization approaches for intrinsic image decomposition and shape-from-shading.

a) b) c)

Figure 2.10 Figure from Prados (2006) a) The crater illusion Pentland (1984): From the image we perceive two craters, a small and a big one. But we can turn these craters into volcanoes (although upside down) if we imagine the light source to be at the bottom of the picture rather than at the top. This picture is actually that of a pair of ash cones in the Hawaiian Island, not that of a pair of craters. b-c) "Bas-relief Ambiguity" Belhumeur et al. (1999): Frontal and side views of a marble bas-relief sculpture. Notice how the frontal views appear to have full 3-dimensional depth, while the side view reveals the flattening. This demonstrates that the image b) can be produced by two surfaces: the three-dimensional surface we imagine by visualizing image b) and the actual bas-relief which is at the origin of the two photos b) and c).

Retinex theory of lightness constancy Land and McCann (1971) is one of earliest priors to solve the intrinsic image problem. It states that low-gradients in an image can be explained by shading variation while high gradient are texture variations from the reflectance image. Following this insight, the reflectance image is assumed to be piecewise constant. In practice, this assumption can be modelled by computing the likelyhood of neighbouring-pixel variations under assumption of a heavy-tail distribution (the variation should be small and sparse).

Parsimony of reflectance Gehler et al. (2011); Omer and Werman (2004) is another useful prior to solve the intrinsic image problem which complements the Retinex prior. The assumption is that the reflectance image only has a few discrete colors. In practice, the entropy of the reflectance image is minimized to obtain a sparse palette of colors.

Color prior of reflectance Barron and Malik (2015) states that some colors are more likely than others in the reflectance image. White-balance or autocontrast algorithms rely on similar priors: the white-world assumption penalizes reflectance for being non-white, the gray-world assumption penalizes reflectance for being non-gray.

Shape Smoothness Jinggang Huang et al. (2000); Woodford et al. (2008) is a standard assumption in Shape from Shading algorithm. 3D shapes tend to bend rarely which can be modeled by minimizing the variation of mean curvature. This can be viewed as the extension of Retinex theory to 3D shapes which makes sense since image statistics are the projection of 3D shapes statistics.

Shape contours Brady and Yuille (1984); Koenderink (1984); Mamassian et al. (1996) are also useful in Shape from Shading algorithm. The normal at an image contour point is orthogonal to the view vector and the tangent plane of the contour.

These cues are at the core of the first optimization approaches to single-image reconstruction. More recently, Zhou et al. (2015) learn priors on reflectance with deep networks, while Li and Snavely (2018); Ma et al. (2018); Takuya Narihira and Yu (2015) use Convolutional Neural Networks to directly infer an intrinsic image decomposition directly from an input image.

2.3.1.2 Learning depth prediction

Several approaches avoid an explicit decomposition in a shading and reflectance image and directly infer a depth map from a single image by learning to solve this task on a data collection. In this section, we present an overview of such methods doing "monocular depth estimation". An important aspect of the latest approaches in this category is that they work on real outdoor images, and full 3D scenes. They do not however reconstruct the unseen parts of a scene. In contrast, template-based methods and deep single-view object reconstructions methods discussed in the next sections only work at the object level, but reconstruct the full object including the unseen parts.

In their seminal approach called Make3D, Saxena et al. (2009) proposed to group pixels in the input image into super-pixels and explain each super-pixel by a local plane in 3D. They cast this as a supervised learning problem and train an linear model to predict the orientation and 3D location of each plane using a dataset of laser scans. Since global image context is a useful cue for depth prediction, the predictions are post-processed by a Markov Random Field.

More recently, Eigen et al. (2014) use Convolutional Neural Networks taking directly as input a single image and predicting a depth map, and train them with a fully supervised regression loss function. Eigen and Fergus (2015) show that a multi-task training objective including semantic segmentation of the image helps for depth estimation. Building on this approach, Bo Li et al. (2015) propose to post-process the predicting depth maps with Conditional Random Fields.

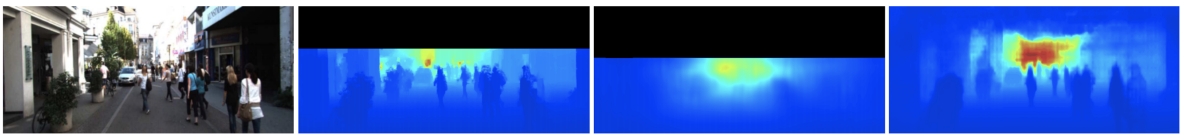


Figure 2.11 Figure from Fu et al. (2018). Single-image Depth Prediction on KITTI Geiger et al. (2013). Eigen et al. (2014) cast depth estimation as a regression problem while Fu et al. (2018) propose DORN, an ordinal classification approach.

Instead of a regression problem, Su et al. (2019) propose to cast depth estimation as a classification problem by discretizing the range of depth predictions. A standard classification loss like the cross-entropy ignores the order that class labels might have and penalizes equally a wrong classification regardless of the class. This is not well suited to quantized depth values which have a clear order, so Fu et al. (2018) use an ordinal classification loss yielding long-standing state-of-the-art results, illustrated in Figure 2.11.

Godard et al. (2017) propose to learn on stereo data without depth supervision. In their approach, given the left image of a calibrated stereo pair, a neural network predicts the disparity maps between the left and right image, which are later used to reconstruct the stereo pair by sampling. In contrast to other approaches, this network is training with an unsupervised reconstruction loss on binocular stereo footages, and yields competitive performance.

2.3.2 Template alignment methods

The seminal work of Lawrence G. Robert introduces a different line of approaches using shape templates to reconstruct the full shape of a object or a scene from a single image Roberts (1963). In his own words: "we shall assume that the objects seen could be constructed out of parts with which we are familiar. That is, either the whole object is a transformation of a preconceived model, or else it can be broken into parts that are. ... The only requirement is that we have a complete description of the three-dimensional structure of each model.". We start the discussion with methods using arrangement of simple templates like cuboids, then discuss methods that suppose access to a predefined 3D model of the full shape and finally discuss methods using more complicated morphable templates.

2.3.2.1 Alignment of simple geometric templates

Early approaches used a simplification of the world, called the blocks world, in which 3D shapes are polyhedral in a uniform background. This assumption enables using projective image formation models: 3d lines map to 2d lines and polyhedral faces to polygons. Using

this assumption, Roberts (1963) proposes a computational approach to SVR that first extracts lines in an image, then matches the projected 3D lines of polyhedrals to the extracted lines. A practical applications based on the blocks world assumption was the MIT robot reconstructing block structures from an input image in a constrained environment. The reader interested in early successes of computer vision can see a video of the demo at the address: http://projects.csail.mit.edu/films/aifilms/digitalFilms/9mp4/88-eye.mp4.

Despite these results, it became clear that the blocks world assumption does not hold in real world scenes. In a effort to use a less restrictive assumption of 3D objects, Thomas Binford proposes to use Generalized Cylinders templates Binford (1971). Generalized Cylinders can be seen as sweeps of a cross-section along a curved axis. Another classical family of templates are Superquadrics Pentland (1986). Superquadrics are parametric family of shapes, which include cubes, octahedra, cylinders, lozenges and spindles.

These ideas had a strong echo in the psycho-physics community. Irving Biederman proposed a model of the human visual system based on morphable Generalized Cylinders called geons. His theory presented in Biederman (1985, 1987), called recognition-by-components, suggests that our visual system recognizes object by separating them into arrangement of geons, and thata set of 36 types of geons are enough to describe most daily objects. A cup would for instance be split in two geons: a cylinder and a handle. Our perception of a specific geons arrangement would then be compared with past observations to recognize the object. Biederman supports his theory with an analog theory on the composition of speech stating that a combination of 55 phonemes can make up any word in any language.

At the time, approaches based on blocks, Generalized Cylinders or Superquadrics required considerable hand-crafting and constrained settings to reconstruct 3D shapes from 2D views. The drastic improvement brought by machine learning can be understood in light of Pentland (1986) quote: "The computation of such a depth map has been the major focus of effort in vision research over the last decade and, although the final results are not in, the betting is that such depth maps are impossible to obtain in the general, unconstrained situation."

Reignited by the seminal work of Tulsiani et al. (2016) three decades later, several approaches use deep learning to learn how to reconstruct 3D object with arrangement of simple templates. Tulsiani et al. (2016) propose to let a neural network predict a set of parameter for cuboids given an input image. These parameters include rotation, translation, scaling and occurrence of each cuboid in a sampled object from the data collection. The network is trained with a global reconstruction loss on a collection of pairs 2D image/3D shape. Instead of cuboids, Paschalidou et al. (2020, 2019) use Superquadrics and predict their shape, position and occurrence with a neural network.

Figure 2.12 Primitive-based reconstructions. Figure from Tulsiani et al. (2016) and Paschalidou et al. (2019). Primitive reconstructions are structure, parsimonious and provide part correspondence across shapes.

Predicting part occurrence in a shape is a hard challenge for these methods because it is non-differentiable. Tulsiani et al. (2016) propose to use reinforcement learning to learn an occurrence probability for each primitive which makes their learning procedure is unstable. Paschalidou et al. (2019) cast the objective as a supervised learning problem using a astute mathematical reformulation of the problem. While this leads to stable training, primitives tend to specialize to regions of space rather than structural functions. On the applications side, Tulsiani et al. (2016) demonstrate that primitive-based reconstructions can be consistent across different instances without explicit supervision. Figure 2.16 shows qualitative examples of recent methods for primitive-based reconstruction. Paschalidou et al. (2020) subdivide recursively the shape in Superquadrics yielding a hierarchical structure, and predict an estimate of the quality of the reconstruction used as a criterion to stop the recursive subdivision.

Beside single-image shape reconstruction, Tulsiani et al. (2016) demonstrate that primitive-based reconstructions can be consistent across different instances without explicit supervision. Figure 2.16 shows qualitative examples of deep-learning based methods aligning simple templates for reconstruction.

2.3.2.2 Alignment by recognition

Another approach is to apply image classification methods to select an "example" from a collection of 3D shapes, then align that "example" to the input image. This naturally gives a prior for unseen parts in the image but generalize poorly outside of the set of examples, and requires the entire collection to be available at test time. This tasks is called Category-level 2D-3D alignment and generalizes Instance-level 2D-3D alignment dating back to Roberts's PhD, which assumes prior knowledge of the 3D model and only focuses on alignment.

Previous methods rely on bag-of-features for classification Csurka et al. (2004); Opelt et al. (2004) and use image cues for alignment like contours Mundy (2006); Russell et al. (2011), or local features Rothganger et al. (2006). Modern approaches based on Convolutional Neural Networks are the state of the art at image classification and object detection since 2012 Girshick et al. (2014); He et al. (2016b); Krizhevsky et al. (2012). Similarly for 3D object alignment, also called 6D pose estimation, neural network based approaches achieve the state of the art Li et al. (2018b); Xiao et al. (2019).

2.3.2.3 Human morphable template

Instead of simple templates like cuboids, several approaches use complex morphable templates to reconstruct 3D shapes, and in particular 3D shapes of humans. There is a strong focus on single-view reconstruction of human shapes because of commercial applications. Creating personalized 3D avatars has applications ranging from video games, virtual try-on of clothes, to healthcare management. Automatic human reconstruction is also likely to play a key role in action recognition Weinzaepfel and Rogez (2019). Indeed current action recognition systems tend to rely on image context but other elements are likely key to predict human intent, such as hand, face and body poses. In this thesis, we also place a strong emphasis on human reconstruction, especially in Chapter 4 for shape matching. We start by discussing single-part models describing only the hand or the face, then morphable templates of the full human body.

Morphable templates of the human body parts. Earlier reconstruction approaches split the human body to simplify the problem and focus on hands and faces. We refer the reader to Brunton et al. (2014); Zollhöfer et al. (2018) for a survey on face reconstruction and to Yuan et al. (2018) for a survey on hand reconstruction. Since the pioneering method of Blanz and Vetter (1999), many approaches tackle face reconstruction by blending a PCA decomposition of faces Ekman and Friesen (1978). More recently, Hasson et al. (2019a) leverage MANO, a hand morphable template following SMPL formulation of the human body Romero et al. (2017). In their approach, a neural networks predicts the parameters of the MANO model from an image. Note that Hasson et al. (2019a) also use our single-image object reconstruction approach from Chapter 3 to jointly reconstruct manipulated objects.

Holistic human templates. Most related to our work in Chapter 3 and Chapter 4 are methods that reconstruct humans from a single image using a mophable template of the full human body. Interestingly, Pavlakos et al. (2019) shows that holistic approaches reconstructing the full body perform better at hand reconstruction than approaches specialized for hands.To estimate a full body shape, the reference morphable template is SMPL Loper et al. (2015). Many deep

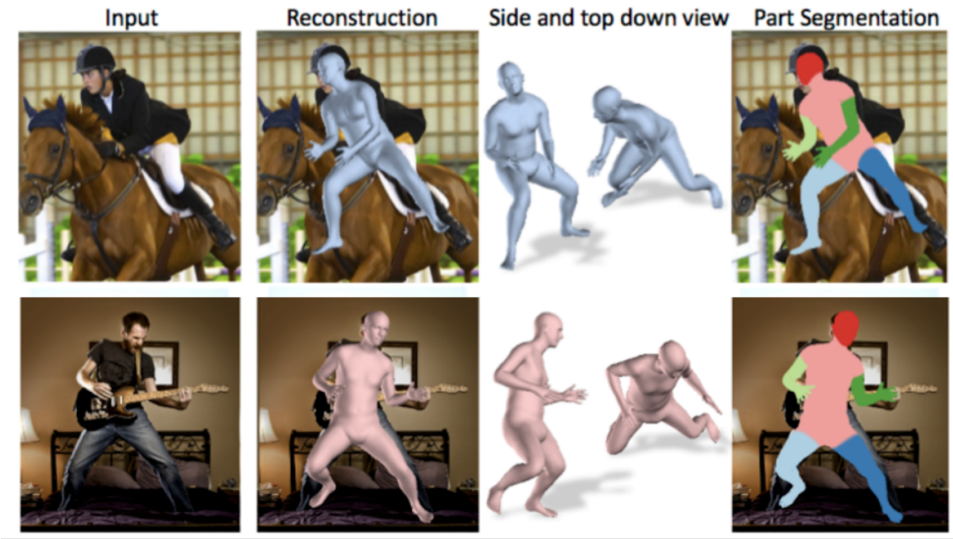


Figure 2.13 Figure from Kanazawa et al. (2018a). Real-time single-view reconstruction of humans using a morphable template Loper et al. (2015). The method infers the full 3D body even in case of occlusions and truncations.

approaches use neural networks to estimate SMPL parameters from an image. The two key challenges are that: there is a lack a 3D annotated pairs (image, corresponding 3D model), and many SMPL parameters can explain the same image.

Faced with the lack of annotated data for human reconstruction, Bogo et al. (2016) use as input a pre-trained network trained for single-image 2D-joint estimation for which there is training data. They train a neural network to estimate SMPL parameters and ensure that the projected 3D keypoints match the 2D keypoints. Instead of relying on 2D key-joint estimation, Varol et al. (2018) directly fit the SMPL model to an image and supervise the model with a reprojection loss. Even if no large-scale annotated data exist for human template fitting, they leverage all 3D annotations available in a multi-task framework. In addition to the reprojection loss, their model is trained for 2D segmentation, 2D pose predictions, and 3D pose predictions.

Since many pose and shape parameters can explain the same image, Pavlakos et al. (2019) discard unrealistic poses that interpenetrate, Bogo et al. (2016) learn a shape prior with a Variational AutoEncoder (VAE) to encourage realistic pose and shapes parameters. Instead of a VAE, Kanazawa et al. (2018a) train a GAN discriminator which tells whether the parameters are from a real pose or not. Figure 2.13 illustrate template-fitting examples using their approach.

Approaches that use morphable template have improved a lot recently and now work on in-the-wild data Kanazawa et al. (2018a). The main limitation of these approaches is that they assume the existence of a morphable template for the reconstructed category. This assumption

holds for humans and some animals Zuffi et al. (2018, 2017), but do not hold for arbitrary object categories.

2.3.3 Deep learning for single-image reconstruction of arbitrary objects

Leveraging morphable template enable accurate reconstructions of 3D shape, but such template are hard to create and only exist for a few object categories. In this section, we discuss the deep learning approaches most related to this thesis that predict the full 3D shape of arbitrary objects.

Deep SVR approaches are typically structured in an autoencoder architecture. This splits the task into two implicit sub-objectives. The encoder network is tasked to extract high-level information from the input image in the form of an embedding vector (typically in $\mathbb{R}^{1024}$). The decoder use this embedding to generates a 3D shape. This decoupling is convenient since encoders and decoders architectures can be easily switched.

Most deep approaches, including this thesis, use the ResNet architecture from He et al. (2016a) to encode images. The decoder network and especially the choice of data representation is the main discriminative factor among deep SVR methods. In the following we briefly detail decoder architectures for SVR in volumetric representation Choy et al. (2016); Mescheder et al. (2019), point clouds Fan et al. (2017), and meshes Wang et al. (2018a). In Chapter 3, we introduce a decoder architecture that learn parametric surface deformation and can generate a mesh. Note that this has been a very active field of research in the last years and many of the methods we discuss in this section have been developed during the PhD work presented here.

2.3.3.1 Volumetric representations

One of the earliest shape representation for deep SVR is the voxel representation. A voxel grid is a 3D regular grid which regularly subdivides a bounding box in the 3D space. Voxels are the natural extension of pixels in 3D. Their value is 1 if there the voxel is inside the shape and otherwise. This representation can be defined for any topology but assumes a clear notion of interior and exterior, which is not the case for thin parts like the sails of a boat.

The structural similarity between voxels and pixels allows direct generalization of 2D operators. Choy et al. (2016) propose a decoder architecture which is simply a stack of 3D convolutional filters and non-linearities. This influential idea was reused to reconstruct voxel grids from various input types like several images Choy et al. (2016); Girdhar et al. (2016), 3D objects with missing shape parts Han et al. (2017); Wu et al. (2015) and sketches Delanoy et al. (2018). The bottleneck of such direct volumetric representation is memory consumption. It scales cubicaly with the resolution which limits the reconstructions to coarse resolutions.

Figure 2.14 Volumetric reconstructions. Figure from Xu et al. (2019). Qualitative reconstructions from real-world examples (input image and two views of the 3D reconstruction). In contrast, our method in chapter 3 learns parametric surface deformations.

To overcome this memory issue, subsequent work has optimized the grid storage with octrees Häne et al. (2017); Riegler et al. (2017); Tatarchenko et al. (2017). Also to adress the memory issue, Li et al. (2017) split the object in parts and generate a voxel representations of each part instead instead of the bounding box.

Recently, Chen and Zhang (2019); Mescheder et al. (2019); Park et al. (2019a) introduced a line of work closely related to the method developed in chapter 3. They propose to let a neural network learn a piece-wise linear approximation of the continuous occupancy function. Chen and Zhang (2019); Park et al. (2019a) approximate the signed distance function of an object instead of the occupancy grid. The signed distance function gives analytical access to surface normals. Xu et al. (2019) feed this new representation with global and local image features and achieve a big qualitative step forward in single-view reconstruction of volumetric representations. These methods and our method in chapter 3 share the same insight: letting a neural network learn piece-wise linear approximation of a continuous function.

2.3.3.2 Point-Cloud representation.

Instead of modeling volumes, Fan et al. (2017) propose to reconstruct point clouds with neural networks. In a simplified version of their work, to generate a point cloud with N points, an embedding is extracted from an input image and later fed to a Multi-Layered Perceptron (MLP) which outputs N 3-channel neurons. Each neurons then abstract a 3D point. To train the autoencoder, they compare the Chamfer distance or the Earth-Mover distance Rubner et al. (2000). While the Chamfer distance is easy to compute since it is an embarrassingly parallel

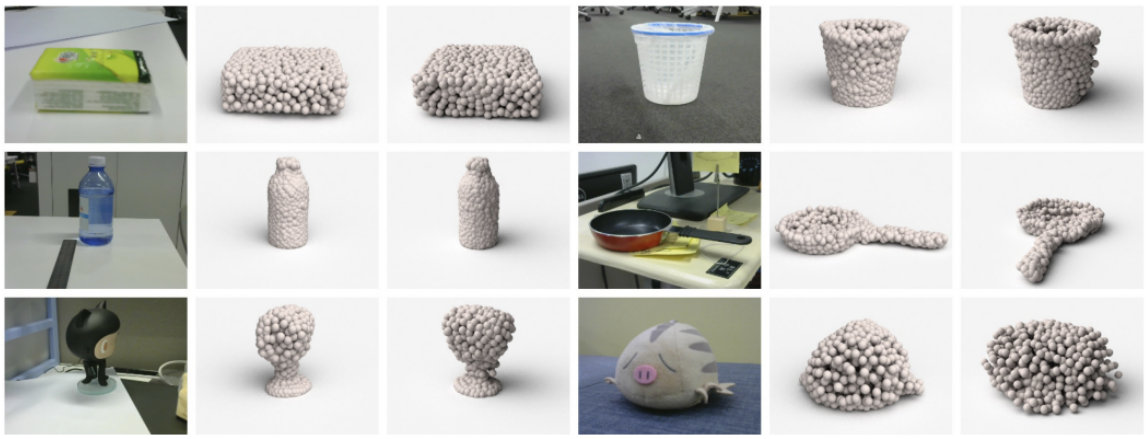


Figure 2.15 Point-based reconstructions. Figure from Fan et al. (2017). Qualitative reconstructions from real-world examples (input image and two views of the 3D reconstruction). In contrast, our method in chapter 3 generates meshes.

nearest-neighbor based loss, the earth-mover distance computes the optimal transport plan between the two pointclouds. In this thesis, we also train our deep networks with the Chamfer Distance.

A limitation of reconstructing point clouds is the lack of surface connectivity (e.g., triangular surface tessellation), as shown in Figure 2.15. In contrast, in Chapter 3, our new representation can output directly a mesh.

2.3.3.3 Mesh

Building on Fan et al. (2017), we introduce in Chapter 3, concurrently to Wang et al. (2018a), the first neural mesh decoder, called AtlasNet. AtlasNet, Pixel2mesh Wang et al. (2018a) and Pixel2mesh++ Wen et al. (2019) are based on the deformation of a template surface. In contrast to volumetric approaches, they do not model topological variation well but are well-suited to model thin-structures. To perform surface deformations of a template mesh into a target mesh, Pixel2mesh and Pixel2mesh++ rely on graph-CNN for non-euclidean manifold while we approximate surface deformations with piece-wise linear function encoded in an MLP. We refer the reader to Wu et al. (2019) for a survey on graph-CNN for non-euclidean surfaces.

Interestingly, the idea of deforming surfaces through MLPs predicting a piece-wise linear approximation of the deformation concurrently emerges in the work of Yang et al. (2018). While we apply surface deformation to do single-image reconstruction of meshes, Yang et al. (2018) use this representation for self-supervised feature learning. More precisely, they train an autoencoder on 3D shapes and show that the latent features are discriminative since a linear

Figure 2.16 Mesh-based reconstructions. Figure from Wang et al. (2018a). Qualitative reconstructions from real-world examples. Results for our method can be found in chapter 3.

SVM get high classification score with them.

Figure 2.17 compares the latest deep approaches discussed in this section. While there has been significant progress in single-view reconstruction in the past years, the main challenges ahead are now to make it work on real world data, scale the current results to full scenes, and solve the original intrinsic image problem: reconstructing more than plain geometry but also texture, material properties and illumination (light-fields).

We conclude this section with an interesting discussion initiated by Tatarchenko et al. (2019) on what do deep autoencoders doing single-view reconstruction actually learn. It is hard to tell how trained neural networks infer a 3D shape from an image since the whole process is implicit. The thesis of Tatarchenko et al. (2019) is that they tend more to solve a recognition problem rather than a reconstruction problem. In other words, they would typically use global information on the image to select an instance from the training set that looks similar to the input, but would not rely on local low-level image cues to reconstruct corresponding local 3D geometry.

Two recognition baselines are proposed to back this claim. The first baseline use k-means on 3D shapes from ShapeNet and train a classification convolutional neural network to predict for each train image its cluster center assignment. The second baseline perform shape retrieval: a CNN is trained to align image embeddings with shape embeddings (obtained with Multi Dimensional Scaling), then shape are selected via nearest-neighbors in the embedding space. Both simple recognition baselines outperform deep approaches, which suggests that deep

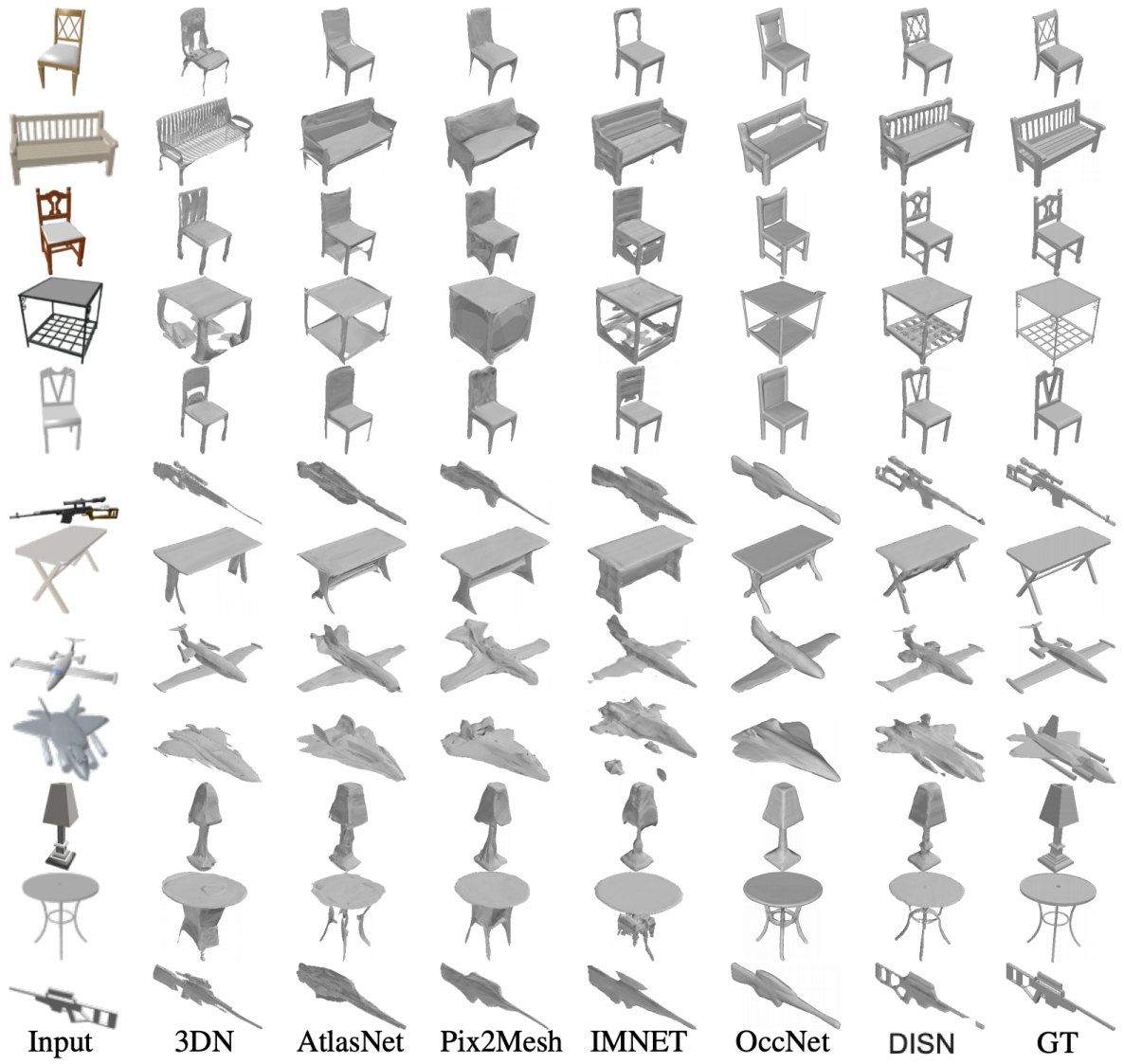


Figure 2.17 Deep single-view reconstruction methods comparison. Figure from Xu et al. (2019). 3DN Wang et al. (2019a), AtlasNet (Chapter 3), Pix2Mesh Wang et al. (2018a) are based on surface deformation while IMNET Chen and Zhang (2019), OccNet Mescheder et al. (2019), DISN Xu et al. (2019) predict volumetric functions.

auto-encoder may also learn recognition under the hood.

Chapter 3

AtlasNet: A Papier-Mache Approach to Learning 3D Surface Generation

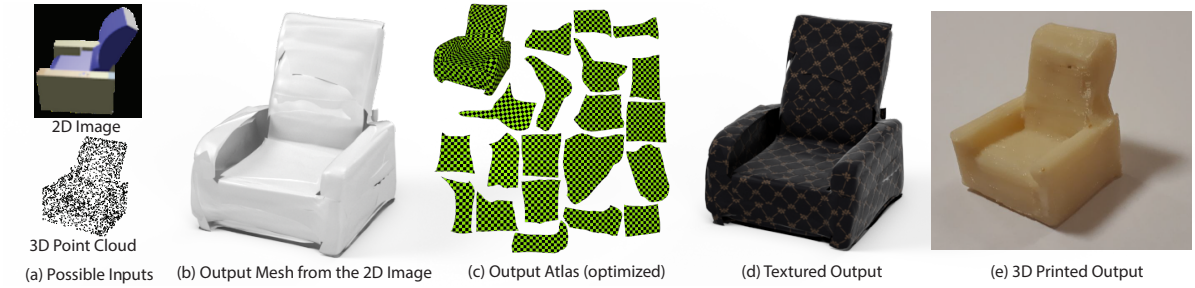


Figure 3.1 Given input as either a 2D image or a 3D point cloud (a), we automatically generate a corresponding 3D mesh (b) and its atlas parameterization (c). We can use the recovered mesh and atlas to apply texture to the output shape (d) as well as 3D print the results (e).

Abstract

In this chapter, we introduce the key idea behind the line of work presented in this thesis: representing 3D surfaces by their deformation of a template and modelling these parametric deformations with neural networks. In contrast to deep methods generating voxel grids or point clouds, our approach naturally infers a surface representation of the shape. Beyond its novelty, our new shape generation framework, AtlasNet, comes with significant advantages, such as improved precision and generalization capabilities, and the possibility to generate a shape of arbitrary resolution without memory issues. We demonstrate these benefits and compare to strong baselines on the ShapeNet benchmark for two applications: (i) auto-encoding shapes, and (ii) single-view reconstruction from a still image. We also provide results showing its potential for other applications, such as morphing, parametrization, super-resolution, matching, and co-segmentation.

The work presented in this chapter was initially presented in:

"AtlasNet: A Papier-Mache Approach to Learning 3D Surface Generation", Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*.

3.1 Introduction

Significant progress has been made on learning good representations for images, allowing impressive applications in image generation Isola et al. (2017); Zhu et al. (2017). However, learning a representation for generating high-resolution 3D shapes remains an open challenge. Representing a shape as a volumetric function Choy et al. (2016); Häne et al. (2017); Tatarchenko et al. (2017) only provides voxel-scale sampling of the underlying smooth and continuous surface. In contrast, a point cloud Qi et al. (2017a,b) provides a representation for generating on-surface details Fan et al. (2017), efficiently leveraging sparsity of the data. However, points do not directly represent neighborhood information, making it difficult to approximate the smooth low-dimensional manifold structure with high fidelity.

To remedy shortcomings of these representations, surfaces are a popular choice in geometric modeling. A surface is commonly modeled by a polygonal mesh: a set of vertices, and a list of triangular or quad primitives composed of these vertices, providing piecewise planar approximation to the smooth manifold. Each mesh vertex contains a 3D (XYZ) coordinate, and, frequently, a 2D (UV) embedding to a plane. The UV parameterization of the surface provides an effective way to store and sample functions on surfaces, such as normals, additional geometric details, textures, and other reflective properties such as BRDF and ambient occlusion. One can imagine converting point clouds or volumetric functions produced with existing learned generative models as a simple post-process. However, this requires solving two fundamental, difficult, and long-standing challenges in geometry processing: global surface parameterization and meshing.

In this chapter, we explore learning the surface representation directly. Inspired by the formal definition of a surface as a topological space that locally resembles the Euclidean plane, we seek to approximate the target surface locally by mapping a set of squares to the surface of the 3D shape. The use of multiple such squares allows us to model complex surfaces with non-disk topology. Our representation of a shape is thus extremely similar to an atlas, as we will discuss in Section 3.3. The key strength of our method is that it jointly learns a parameterization and an embedding of a shape. This helps in two directions. First, by ensuring that our 3D points come from 2D squares we favor learning a continuous and smooth 2-manifold structure. Second, by generating a UV parameterization for each 3D point, we generate a global surface parameterization, which is key to many applications such as texture mapping and surface meshing. Indeed, to generate the mesh, we simply transfer a regular mesh from our 2D squares to the 3D surface, and to generate a regular texture atlas, we simply optimize the metric of the square to become as-isometric-as-possible to the corresponding 3D shape (Fig. 3.1).

Since our work deforms primitive surface elements into a 3D shape, it can be seen as bridging the gap between the recent works that learn to represent 3D shapes as a set of simple primitives, with a fixed, low number of parameters Tulsiani et al. (2016) and those that represent 3D shapes as an unstructured set of points Fan et al. (2017). It can also be interpreted as learning a factored representation of a surface, where a point on the shape is represented jointly by a vector encoding the shape structure and a vector encoding its position. Finally, it can be seen as an attempt to bring to 3D the power of convolutional approaches for generating 2D images Isola et al. (2017); Zhu et al. (2017) by sharing the network parameters for parts of the surface.

Our contributions. In this chapter:

- We propose a novel approach to 3D surface generation, dubbed *AtlasNet*, which is composed of a union of learnable parametrizations. These learnable parametrizations transform a set of 2D squares to the surface, covering it in a way similar to placing strips of chapter on a shape to form a papier-mâché. The parameters of the transformations come both from the learned weights of a neural network and a learned representation of the shape.
- We show that the learned parametric transformation maps locally everywhere to a surface, naturally adapts to its underlying complexity, can be sampled at any desired resolution, and allows for the transfer of a tessellation or texture map to the generated surface.
- We demonstrate the advantages of our approach both qualitatively and quantitatively on high resolution surface generation from (potentially low resolution) point clouds and 2D images
- We demonstrate the potential of our method for several applications, including shape interpolation, parameterization, and shape collections alignment.

All the code is available at the project webpage[1].

3.2 Learning representations for 2-manifolds

3D shape analysis and generation has a long history in computer vision. We already discussed several approaches in Section 2.3, in particular 3D shape generation using deep networks. In this section, we only discuss the most directly related works for representation learning for 2-manifolds A polygon mesh is a widely-used representation for the 2-manifold surface of 3D shapes. Establishing a connection between the surface of the 3D shape and a 2D domain, or surface parameterization, is a long-standing problem in geometry processing, with applications

[1] https://github.com/ThibaultGROUEIX/AtlasNet.

in texture mapping, re-meshing, and shape correspondence Hormann et al. (2008). Various related representations have been used for applying neural networks on surfaces. The geometry image representation Gu et al. (2002); Sander et al. (2003) views 3D shapes as functions (e.g., vertex positions) embedded in a 2D domain, providing a natural input for 2D neural networks Sinha et al. (2016). Various other parameterization techniques, such as local polar coordinates Boscaini et al. (2016a); Masci et al. (2015) and global seamless maps Maron et al. (2017) have been used for deep learning on 2-manifolds.

Unlike these methods, we do not need our input data to be parameterized. Instead, we learn the parameterization directly from point clouds. Moreover, these methods assume that the training and testing data are 2-manifold meshes, and thus cannot easily be used for surface reconstructions from point clouds or images.

3.3 Locally parameterized surface generation

In this section, we detail the theoretical motivation for our approach and present some theoretical guarantees.

We seek to learn to generate a surface of a 3D shape. A subset $\mathcal{S}$ of $\mathbb{R}^3$ is a *2-manifold* if, for every point $\mathbf{p} \in \mathcal{S}$, there is an open set U in $\mathbb{R}^2$ and an open set W in $\mathbb{R}^3$ containing $\mathbf{p}$ such that $\mathcal{S} \cap W$ is homeomorphic to U. The set homeomorphism from $\mathcal{S} \cap W$ to U is called a *chart*, and its inverse a *parameterization*. A set of charts such that their images cover the 2-manifold is called an *atlas* of the 2-manifold. The ability to learn an atlas for a 2-manifold would allow a number of applications, such as transfer of a tessellation to the 2-manifold for meshing and texture mapping (via texture atlases). In this paper, we use the word *surface* in a slightly more generic sense than *2-manifold*, allowing for self-intersections and disjoint sets.

We consider a local parameterization of a 2-manifold and explain how we learn to approximate it. More precisely, let us consider a 2-manifold $\mathcal{S}$, a point $\mathbf{p} \in \mathcal{S}$ and a parameterization φ of $\mathcal{S}$ in a local neighborhood of $\mathbf{p}$. We can assume that φ is defined on the open unit square $]0,1[^2$ by first restricting φ to an open neighborhood of $\varphi^{-1}(\mathbf{p})$ with disk topology where it is defined (which is possible because φ is continuous) and then mapping this neighborhood to the unit square.

We pose the problem of learning to generate the local 2-manifold previously defined as one of finding a parameterizations $\varphi_\theta(x)$ with parameters θ which map the open unit 2D square $]0,1[^2$ to a good approximation of the desired 2-manifold $\mathcal{S}_{\text{loc}}$. Specifically, calling $\mathcal{S}_\theta = \varphi_\theta(]0,1[^2)$, we seek to find parameters θ minimizing the following objective function,

$$\min_\theta \mathcal{L}(\mathcal{S}_\theta, \mathcal{S}_{\text{loc}}) + \lambda \mathcal{R}(\theta), \tag{3.1}$$

where $\mathcal{L}$ is a loss over 2-manifolds, $\mathcal{R}$ is a regularization function over parameters θ, and λ is a scalar weight. In practice, instead of optimizing a loss over 2-manifolds $\mathcal{L}$, we optimize a loss over point sets sampled from these 2-manifolds such as Chamfer and Earth-Mover distance.

One question is, how do we represent the functions φ_θ? A good family of functions should (i) generate 2-manifolds and (ii) be able to produce a good approximation of the desired 2-manifolds S_{loc}. We show that multilayer perceptrons (MLPs) with rectified linear unit (ReLU) nonlinearities almost verify these properties, and thus are an adequate family of functions. Since it is difficult to design a family of functions that always generate a 2-manifold, we relax this constraint and consider functions that locally generate a 2-manifold.

Proposition 1. *Let f be a multilayer perceptron with ReLU nonlinearities. There exists a finite set of polygons P_i, $i \in \{1, \ldots, N\}$ such that on each P_i f is an affine function: $\forall x \in P_i,\ f(x) = A_i x + b$, where A_i are 3×2 matrices. If for all i, $\text{rank}(A_i) = 2$, then for any point $\mathbf{p}$ in the interior of one of the P_is there exists a neighborhood $\mathcal{N}$ of $\mathbf{p}$ such that $f(\mathcal{N})$ is a 2-manifold.*

The fact that f is locally affine is a direct consequence of the fact that we use ReLU nonlinearities. If $\text{rank}(A_i) = 2$ the inverse of $A_i x + b$ is well defined on the surface and continuous, thus the image of the interior of each P_i is a 2-manifold.

To draw analogy to texture atlases in computer graphics, we call the local functions we learn to approximate a 2-manifold *learnable parameterizations* and the set of these functions A a *learnable atlas*. Note that in general, an MLP locally defines a rank 2 affine transformation and thus locally generates a 2-manifold, but may not globally as it may intersect or overlap with itself. The second reason to choose MLPs as a family is that they can allow us to approximate any continuous surface.

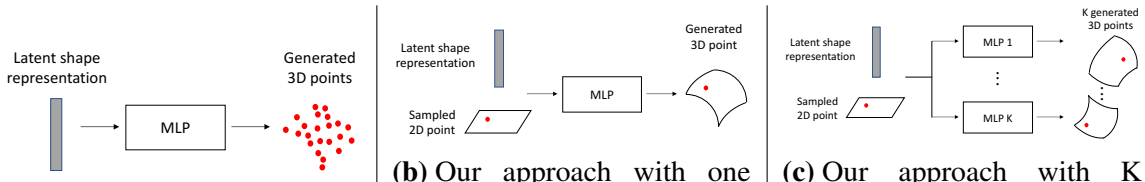
Proposition 2. *Let S be a 2-manifold that can be parameterized on the unit square. For any $\varepsilon > 0$ there exists an integer K such that a multilayer perceptron with ReLU non linearities and K hidden units can approximate S with a precision ε.*

Proof. This is a consequence of the universal representation theorem Hornik (1991) □

In the next section, we show how to train such MLPs to align with a desired surface.

3.4 AtlasNet

In this section we introduce our model, AtlasNet, which decodes a 3D surface given an encoding of a 3D shape. This encoding can come from many different representations such as a point cloud or an image (see Figure 3.2 for examples).



(a) Points baseline.

(b) Our approach with one patch.

(c) Our approach with K patches.

Figure 3.2 Shape generation approaches. All three approaches take as input a latent shape representation (that can be learned jointly with a reconstruction objective) and generate as output a set of points. (a) A baseline deep architecture would simply decode this latent representation into a set of points of a given size. (b) Our approach takes as additional input a 2D point sampled uniformly in the unit square and uses it to generate a single point on the surface. Our output is thus the continuous image of a planar surface. In particular, we can easily infer a mesh of arbitrary resolution on the generated surface elements. (c) This strategy can be repeated multiple times to represent a 3D shape as the union of several surface elements.

3.4.1 Learning to decode a surface

Our goal is, given a feature representation $\mathbf{x}$ for a 3D shape, to generate the surface of the shape. As discussed in Section 3.3, an MLP with ReLUs $\mathcal{D}_\theta$ with parameters θ can locally generate a surface by learning to map points in $\mathbb{R}^2$ to surface points in $\mathbb{R}^3$. To generate a given surface, we need several of these learnable charts to represent a surface. In practice, we consider N learnable parameterizations ϕ_{θ_i} for $i \in \{1, \ldots, N\}$. To train the MLP parameters θ_i, we need to address two questions: (i) how to define the distance between the generated and target surface, and (ii) how to account for the shape feature $\mathbf{x}$ in the MLP? To represent the target surface, we use the fact that, independent of the representation that is available to us, we can sample points on it. Let $\mathcal{A}$ be a set of points sampled in the unit square $[0,1]^2$ and $\mathcal{S}$ a set of points sampled on the target surface. Next, we incorporate the shape feature $\mathbf{x}$ by simply concatenating them with the sampled point coordinates $\mathbf{p} \in \mathcal{A}$ before passing them as input to the MLPs. Our model is illustrated in Figure 3.2b. Notice that the MLPs are not explicitly prevented from encoding the same area of space, but their union should cover the full shape. Our MLPs do depend on the random initialization, but similar to convolutional filter weights the network learns to specialize to different regions in the output without explicit biases. We then minimize the Chamfer loss between the set of generated 3D points and $\mathcal{S}$,

$$\mathcal{L}(\theta) = \sum_{\mathbf{p}\in\mathcal{A}} \sum_{i=1}^{N} \min_{\mathbf{q}\in\mathcal{S}} |\phi_{\theta_i}(\mathbf{p};\mathbf{x}) - \mathbf{q}|^2 + \sum_{\mathbf{q}\in\mathcal{S}} \min_{i\in\{1,\ldots,N\}} \min_{\mathbf{p}\in\mathcal{A}} |\phi_{\theta_i}(\mathbf{p};\mathbf{x}) - \mathbf{q}|^2 . \tag{3.2}$$

The left part of equation 4.3 encourages all the deformed points $\phi_{\theta_i}(\mathbf{p};\mathbf{x})$ sampled from $\mathcal{A}$ to be close to $\mathcal{S}$. On the contrary, the right part of the equation encourages all the deformed points q from $\mathcal{A}$ to be close to deformed points $\phi_{\theta_i}(\mathbf{p};\mathbf{x})$ from $\mathcal{S}$.

3.4.2 Implementation details

We consider two tasks: (i) to auto-encode a 3D shape given an input 3D point cloud, and (ii) to reconstruct a 3D shape given an input RGB image. For the auto-encoder, we used an encoder based on PointNet Qi et al. (2017a), which has proven to be state of the art on point cloud analysis on ShapeNet and ModelNet40 benchmarks. This encoder transforms an input point cloud into a latent vector of dimension $k = 1024$. We experimented with input point clouds of 250 to 2500 points. For images, we used ResNet-18 He et al. (2016b) as our encoder. The architecture of our decoder is 4 fully-connected layers of size 1024, 512, 256, 128 with ReLU non-linearities on the first three layers and tanh on the final output layer. We always train with output point clouds of size 2500 evenly sampled across all of the learned parameterizations – scaling above this size is time-consuming because our implementation of Chamfer loss has a compute cost that is quadratic in the number of input points. We experimented with different basic weight regularization options but did not notice any generalization improvement. Sampling of the learned parameterizations as well as the ground truth point-clouds is repeated at each training step to avoid over-fitting. To train for single-view reconstruction, we obtained the best results by training the encoder and using the decoder from the point cloud autoencoder with fixed parameters. Finally, we noticed that sampling points regularly on a grid on the learned parameterization yields better performance than sampling points randomly. All results used this regular sampling.

3.4.3 Mesh generation

The main advantage of our approach is that during inference, we can easily generate a mesh of the shape.

Propagate the patch-grid edges to the 3D points. The simplest way to generate a mesh of the surface is to transfer a regular mesh on the unit square to 3D, connecting in 3D the images of the points that are connected in 2D. Note that our method allows us to generate such meshes at very high resolution, without facing memory issues, since the points can be processed in batches. We typically use 22500 points. As shown in the results section, such meshes are satisfying, but they can have several drawbacks: they will not be closed, may have small holes between the images of different learned parameterizations, and different patches may overlap.

Generate a highly dense point cloud and use Poisson surface reconstruction (PSR) Kazhdan and Hoppe (2013). To avoid the previously mentioned drawbacks, we can additionally densely sample the surface and use a mesh reconstruction algorithm. We start by generating a surface at a high resolution, as explained above. We then shoot rays at the model from infinity and obtain approximately 100000 points, together with their oriented normals, and then can use a standard oriented cloud reconstruction algorithm such as PSR to produce a triangle mesh. We found that high quality normals as well as high density point clouds are critical to the success of PSR, which are naturally obtained using this method.

Sample points on a closed surface rather than patches. To obtain a closed mesh directly from our method, without requiring the PSR step described above, we can sample the input points from the surface of a 3D sphere instead of a 2D square. The quality of this method depends on how well the underlying surface can be represented by a sphere, which we will explore in Section 3.5.1.

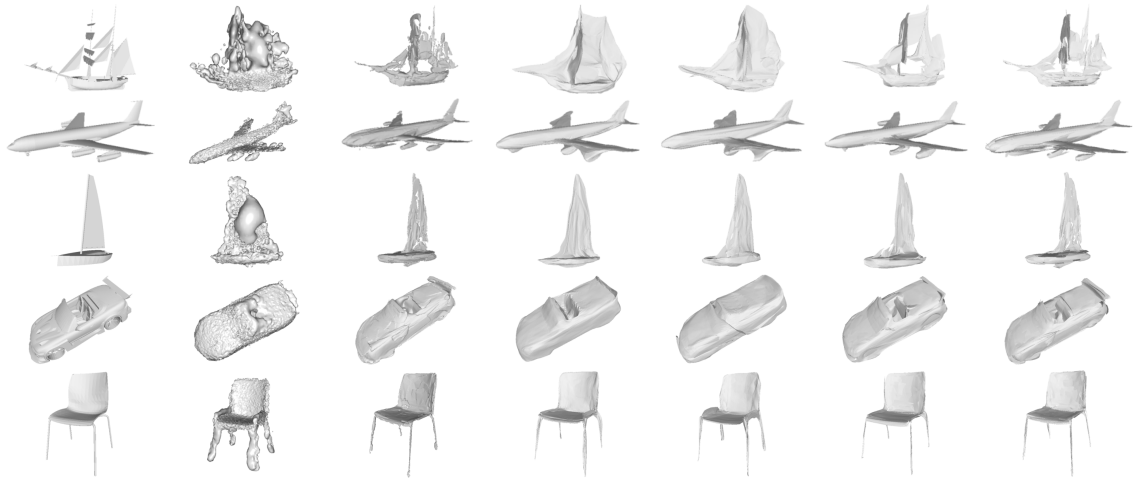
3.5 Results

In this section we show qualitative and quantitative results on the tasks of auto-encoding 3D shapes and single-view reconstruction and compare against several baselines. In addition to these tasks, we also demonstrate several additional applications of our approach.

Data. We evaluated our approach on the standard ShapeNet Core dataset (v2) Chang et al. (2015). The dataset consists of 3D models covering 13 object categories with 1K-10K shapes per category. We used the training and validation split provided by Choy et al. (2016) for our experiments to be comparable with previous approaches. We used the rendered views provided by Choy et al. (2016) and sampled 3D points on the shapes using Wang et al. (2017).

Evaluation criteria. We evaluated our generated shape outputs by comparing to ground truth shapes using two criteria. First, we compared point sets for the output and ground-truth shapes using Chamfer distance ("CD"). While this criteria compares two point sets, it does not take into account the surface/mesh connectivity. To account for mesh connectivity, we compared the output and ground-truth meshes using the "Metro" criteria using the publicly available METRO software Cignoni et al. (1998), which is the average Euclidean distance between the two meshes.

Points baseline. In addition to existing baselines, we compare our approach to the multi-layer perceptron "Points baseline" network shown in Figure 3.2a. The Points baseline network



(a) Ground truth **(b)** Pts baseline **(c)** PSR on ours **(d)** Ours sphere **(e)** Ours 1 **(f)** Ours 25 **(g)** Ours 125

Figure 3.3 Auto-encoder. We compare the original meshes (a) to meshes obtained by running PSR on the point clouds generated by the baseline (b) and on the densely sampled point cloud from our generated mesh (c), and to our method generating a surface from a sphere (d), 1 (e), 25 (f), and 125(g) learnable parameterizations. Notice the fine details in (f) and (g) : e.g. the plane's engine and the jib of the ship.

consists of four fully connected layers with output dimensions of size 1024, 512, 256, 7500 with ReLU non-linearities, batch normalization on the first three layers, and a hyperbolic-tangent non-linearity after the final fully connected layer. The network outputs 2500 3D points and has comparable number of parameters to our method with 25 learned parameterizations. The baseline architecture was designed to be as close as possible to the MLP used in AtlasNet. As the network outputs points and not a mesh, we also trained a second network that outputs 3D points and normals, which are then passed as inputs to Poisson surface reconstruction (PSR) Kazhdan and Hoppe (2013) to generate a mesh ("Points baseline + normals"). The network generates outputs in $\mathbb{R}^6$ representing both the 3D spatial position and normal. We optimized Chamfer loss in this six-dimensional space and normalized the normals to 0.1 length as we found this trade-off between the spatial coordinates and normals in the loss worked best. As density is crucial to PSR quality, we augmented the number of points by sampling 20 points in a small radius in the tangent plane around each point Kazhdan and Hoppe (2013). We noticed significant qualitative and quantitative improvements and the results shown in this paper use this augmentation scheme.

3.5.1 Auto-encoding 3D shapes

In this section we evaluate our approach to generate a shape given an input 3D point cloud and compare against the Points baseline. We evaluate how well our approach can generate the

Method	CD	Metro
Oracle 2500 pts	0.85	1.56
Oracle 125K pts	-	1.26
Points baseline	1.91	-
Points baseline + normals	2.15	1.82 (PSR)
Ours - 1 patch	1.84	1.53
Ours - 1 sphere	1.72	1.52
Ours - 5 patches	1.57	1.48
Ours - 25 patches	1.56	1.47
Ours - 125 patches	**1.51**	**1.41**

Table 3.1 3D reconstruction. Comparison of our approach against a point-generation baseline ("CD" - Chamfer distance, multiplied by 10^3, computed on 2500 points; "Metro" values are multiplied by 10). Note that our approach can be directly evaluated by Metro while the baseline requires performing PSR Kazhdan and Hoppe (2013). These results can be compared with an Oracle sampling points directly from the ground truth 3D shape followed by PSR (top two rows). See text for details.

shape, how it can generalize to object categories not seen during training, and its sensitivity to the number of patches.

Evaluation on surface generation. We report quantitative results for shape generation from point clouds in Table 3.1, where each approach is trained over all ShapeNet categories and results are averaged over all categories. Notice that our approach out-performs the Points baseline on both the Chamfer distance and Metro criteria, even when using a single learned parameterization (patch). Also, the Points baseline + normals has worse Chamfer distance than the Points baseline without normals indicating that predicting the normals decreases the quality of the point cloud generation.

We also report performance for two "oracle" outputs indicating upper bounds in Table 3.1. The first oracle ("Oracle 2500 pts") randomly samples 2500 points+normals from the ground truth shape and applies PSR. The Chamfer distance between the random point set and the ground truth gives an upper bound on performance for point-cloud generation. Notice that our method out-performs the surface generated from the oracle points. The second oracle ("Oracle 125K pts") applies PSR on all 125K points+normals from the ground-truth shape. It is interesting to note that the Metro distance from this result to the ground truth is not far from the one obtained with our method.

We show qualitative comparisons in Figure 3.3. Notice that the PSR from the baseline point clouds (Figure 3.3b) look noisy and lower quality than the meshes produced directly by our

Category		Points baseline	Ours 1 patch	Ours 125 patches
chair	LOO	3.66	3.43	**2.69**
	All	1.88	1.97	**1.55**
car	LOO	3.38	2.96	**2.49**
	All	1.59	2.28	**1.56**
watercraft	LOO	2.90	2.61	**1.81**
	All	1.69	1.69	**1.23**
plane	LOO	6.47	6.15	**3.58**
	All	1.11	1.04	**0.86**

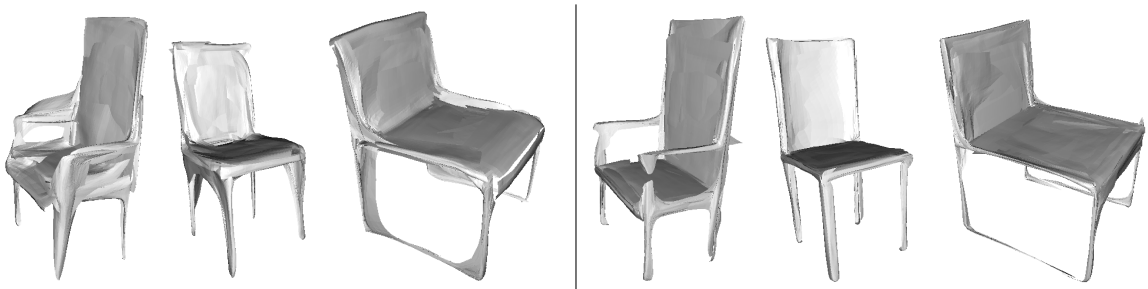
Table 3.2 Generalization across object categories. Comparison of our approach with varying number of patches against the point-generating baseline to generate a specific category when training on all other ShapeNet categories. Chamfer distance is reported, multiplied by 10^3, computed on 2500 points. Notice that our approach with 125 patches out-performs all baselines when generalizing to the new category. For reference, we also show performance when we train over all categories.

method and PSR performed on points generated from our method as described in Section 3.4.3 (Figure 3.3c).

Sensitivity to number of patches. We show in Table 3.1 our approach with varying number of learnable parameterizations (patches) in the atlas. Notice how our approach improves as we increase the number of patches. Moreover, we also compare with the approach described in Section 3.4.3 which samples points on the 3D unit sphere instead of 2D patches to obtain a closed mesh. Notice that sampling from a sphere quantitatively out-performs a single patch, but multiple patches perform better.

We show qualitative results for varying number of learnable parameterizations in Figure 3.3. As suggested by the quantitative results, the visual quality improves with the number of parameterizations. However, more artifacts appear with more parameterizations, such as close-but-disconnected patches (e.g., sail of the sailboat) . We thus used 25 patches for the single-view reconstruction experiments (Section 3.5.2)

Generalization across object categories. An important desired property of a shape auto-encoder is that it generalizes well to categories it has not been trained on. To evaluate this, we trained our method on all categories but one target category ("LOO") for chair, car, watercraft, and plane categories, and evaluated on the held-out category. The corresponding results are reported in Table 3.2 and Figure 3.4. We also include performance when the methods are trained on all of the categories including the target category ("All") for comparison. Notice



(a) Not trained on chairs **(b)** Trained on all categories

Figure 3.4 Generalization. (a) Our method (25 patches) can generate surfaces close to a category never seen during training. It, however, has more artifacts than if it has seen the category during training (b), e.g., thin legs and armrests.

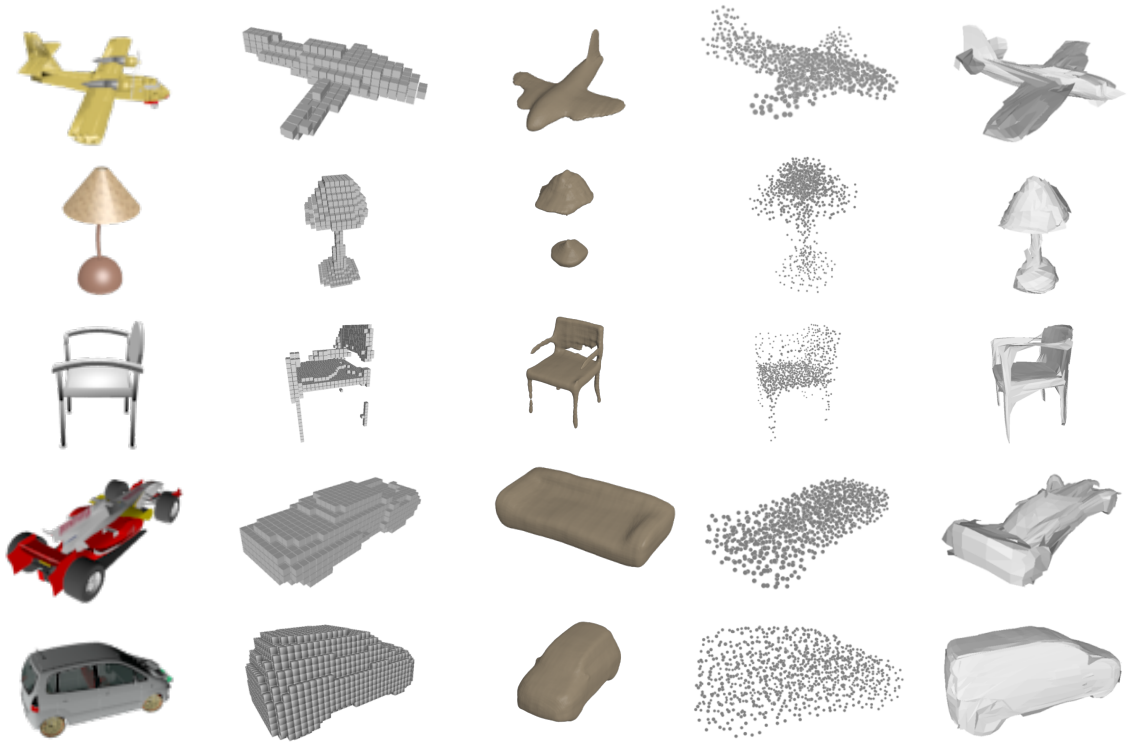
	pla.	ben.	cab.	car	cha.	mon.	lam.	spe.	fir.	cou.	tab.	cel.	wat.	mean
Ba CD	2.91	4.39	6.01	4.45	7.24	**5.95**	7.42	10.4	1.83	**6.65**	4.83	4.66	**4.65**	5.50
PSG CD	3.36	4.31	8.51	8.63	**6.35**	6.47	7.66	15.9	**1.58**	6.92	**3.93**	**3.76**	5.94	6.41
Ours CD	**2.54**	**3.91**	**5.39**	**4.18**	6.77	6.71	**7.24**	**8.18**	1.63	6.76	4.35	3.91	4.91	**5.11**
Ours Metro	1.31	1.89	1.80	2.04	2.11	1.68	2.81	2.39	1.57	1.78	2.28	1.03	1.84	1.89

Table 3.3 Single-View Reconstruction (per category). The mean is taken category-wise. The Chamfer Distance reported is computed on 1024 points, after running ICP alignment with the GT point cloud, and multiplied by 10^3. The Metro distance is multiplied by 10.

that we again out-perform the point-generating baseline on this leave-one-out experiment and that performance improves with more patches. The car category is especially interesting since when trained on all categories the baseline has better results than our method with 1 patch and similar to our method with 125 patches. If not trained on cars, both our approaches clearly outperform the baseline, showing that at least in this case, our approach generalizes better than the baseline. The visual comparison shown Figure 3.4 gives an intuitive understanding of the consequences of not training for a specific category. When not trained on chairs, our method seems to struggle to define clear thin structures, like legs or armrests, especially when they are associated to a change in the topological genus of the surface. This is expected as these types of structures are not often present in the categories the network was trained on.

3.5.2 Single-view reconstruction

We evaluate the potential of our method for single-view reconstruction. We compare qualitatively our results with three state-of-the-art methods, PointSetGen Fan et al. (2017), 3D-R2N2 Choy et al. (2016) and HSP Häne et al. (2017) in Figure 3.5. To perform the comparison for



(a) Input **(b)** 3D-R2N2 **(c)** HSP **(d)** PSG **(e)** Ours

Figure 3.5 Single-view reconstruction comparison. From a 2D RGB image (a), 3D-R2N2 Choy et al. (2016) reconstructs a voxel-based 3D model (b), HSP Häne et al. (2017) reconstructs a octree-based 3D model (c), PointSetGen Fan et al. (2017) a point cloud based 3D model (d), and our AtlasNet a triangular mesh (e).

PointSetGen Fan et al. (2017) and 3D-R2N2 Choy et al. (2016), we used the trained models made available online by the authors. For HSP Häne et al. (2017), we asked the authors to run their method on the images in Fig. 3.5. Note that since their model was trained on images generated with a different renderer, this comparison is not absolutely fair. To remove the bias we also compared our results with HSP on real images for which none of the methods was trained (Fig. 3.6) which also demonstrates the ability of our network to generalize to real images.

Figure 3.5 emphasizes the importance of the type of output (voxels for 3D-N2D2 and HSP, point cloud for PointSetGen, mesh for us) for the visual appearance of the results. Notice the small details visible on our meshes that may be hard to see on the unstructured point cloud or volumetric representation. Also, it is interesting to see that PointSetGen tends to generate points inside the volume of the 3D shape while our result, by construction, generates points on a surface.

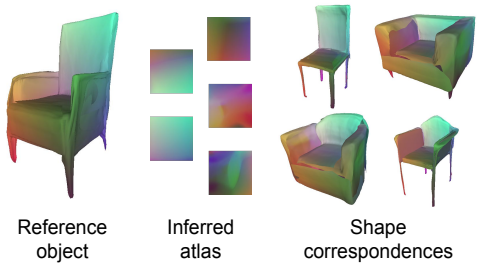
(a) Input **(b)** HSP **(c)** Ours

Figure 3.6 Single-view reconstruction comparison on natural images. From a 2D RGB image taken from internet (a), HSP Häne et al. (2017) reconstructs a octree-based 3D model (b), and our AtlasNet a triangular mesh (c).

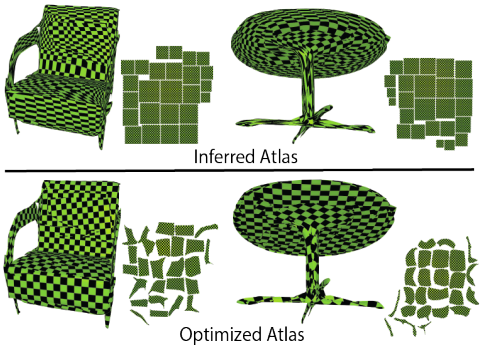
To perform a quantitative comparison against PointSetGen Fan et al. (2017), we evaluated the Chamfer distance between generated points and points from the original mesh for both PointSetGen and our method with 25 learned parameterizations. However, the PointSetGen network was trained with a translated, rotated, and scaled version of ShapeNet with parameters we did not have access to. We thus first had to align the point clouds resulting from PointSetGen to the ShapeNet models used by our algorithm. We randomly selected 260 shapes, 20 from each category, and ran the iterative closest point (ICP) algorithm Besl et al. (1992) to optimize a similarity transform between PointSetGen and the target point cloud. Note that this optimization improves the Chamfer distance between the resulting point clouds, but is not globally convergent. We checked visually that the point clouds from PointSetGen were correctly aligned, and display all alignments on the project webpage[2]. To have a fair comparison we ran the same ICP alignment on our results. In Table 3.3 we compared the resulting Chamfer distance. Our method provides the best results on 6 categories whereas PointSetGen and the baseline are best on 4 and 3 categories, respectively. Our method is better on average and generates point clouds of a quality similar to the state of the art. We also report the Metro distance to the original shape, which is the most meaningful measure for our method.

[2] http://imagine.enpc.fr/~groueixt/atlasnet/PSG.html.

(a) Shape interpolation.



(b) Shape correspondences.



(c) Mesh parameterization.

Figure 3.7 Applications. Results from three applications of our method. See text for details.

To quantitatively compare against HSP Häne et al. (2017), we retrained our method on their publicly available data since train/test splits are different from 3D-R2N2 Choy et al. (2016) and they made their own renderings of ShapeNet data. Results are in Table 3.4. More details are in the supplementary Groueix et al. (2018b).

	Chamfer	Metro
HSP Häne et al. (2017)	11.6	1.49
Ours (25 patches)	**9.52**	**1.09**

Table 3.4 Single-view reconstruction. Quantitative comparison against HSP Häne et al. (2017), a state of the art octree-based method. The average error is reported, on 100 shapes from each category. The Chamfer Distance reported is computed on 10^4 points, and multiplied by 10^3. The Metro distance is multiplied by 10

3.5.3 Additional applications

Shape interpolation. Figure 3.7a shows shape interpolation. Each row shows interpolated shapes generated by our AtlasNet, starting from the shape in the first column to the shape in the

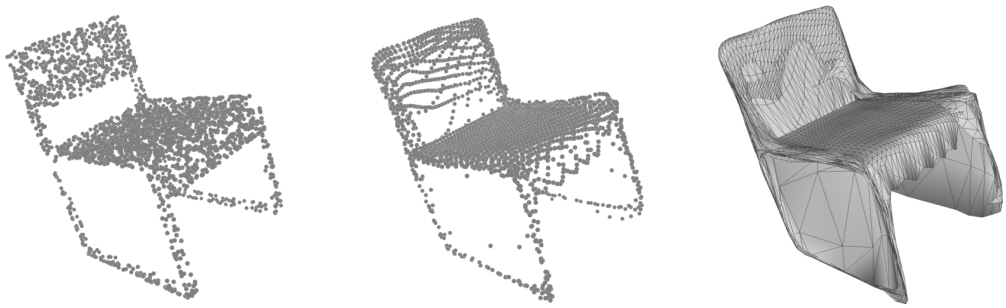
last. Each intermediate shape is generated using a weighted sum of the latent representations of the two extreme shaped. Notice how the interpolated shapes gradually add armrests in the first row, and chair legs in the last.

Finding shape correspondences. Figure 3.7b shows shape correspondences. We colored the surface of reference chair (left) according to its 3D position. We transfer the surface colors from the reference shape to the inferred atlas (middle). Finally, we transfer the atlas colors to other shapes (right) such that points with the same color are parametrized by the same point in the atlas. Notice that we get semantically meaningful correspondences, such as the chair back, seat, and legs without any supervision from the dataset on semantic information.

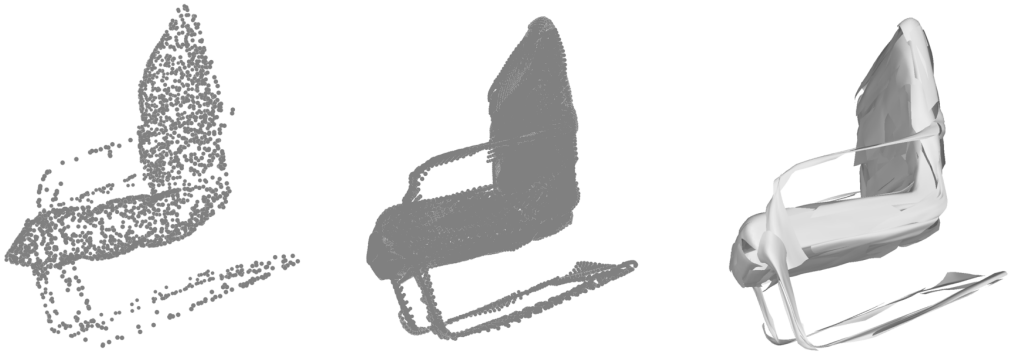
Mesh parameterization Most existing rendering pipelines require an atlas for texturing a shape (Figure 3.7c). A good parameterization should minimize amount of area distortion (E_a) and stretch (E_s) of a UV map. We computed average per-triangle distortions for 20 random shapes from each category and found that our inferred atlas usually has relatively high texture distortion ($E_a = 1.9004, E_s = 6.1613$, where undistorted map has $E_a = E_s = 1$). Our result, however, is well-suited for distortion minimization because all meshes have disk-like topology and inferred map is bijective, making it easy to further minimize distortion with off-the-shelf geometric optimization Kovalsky et al. (2016), yielding small distortion ($E_a = 1.0016, E_s = 1.025$, see bottom row for example).

Limitations and impact We describe two limitations with our approach, illustrated in Figure 3.8. First, when a small number of learned parameterizations are used, the network has to distort them too much to recreate the object. This leads, when we try to recreate a mesh, to small triangles in the learned parameterization space being distorted and become large triangles in 3D covering undesired regions. On the other hand, as the number of learned parameterization increases, results ave visually less pleasing, showing many small separated surface elements overlapping and not stitched together. Bednarík et al. (2020) address some of AtlasNet limitations by introducing different types of regularization, especially conformal, to limit distorsion and a penalty loss that discourage overlapping. The official Github repository for AtlasNet incorporate these ideas in a separate branch.

AtlasNet has also inspired other works from the research community. Two papers in particular propose new ways to use the framework. Williams et al. Williams et al. (2019) leverage the AtlasNet architecture as a deep geometric prior to reconstruct surfaces from noisy point-clouds without learning. Lin et al. Lin et al. (2019) adapted our model to reconstruct 3D meshes from RGB videos. AtlasNet was also used by Hasson et al. (2019b) to jointly reconstruct



(a) Excess of distortion. Notice how, compared to the original point cloud (left), the generated pointcloud (middle) with 1 learned parameterization is valid, but the mapping from squares to surfaces enforces too much distortion leading to error when propagating the grid edges in 3D (right).



(b) Topological issues. Notice how, compared to the original point cloud (left), the generated pointcloud (middle) with 125 learned parameterizations is valid, but the 125 generated surfaces overlap and are not stiched together (right).

Figure 3.8 Limitations. Two main artifacts are highlighted : (a) Excess of distortion when too small a number of learned parameterizations is used, and (b) growing errors in the topology of the reconstructed mesh as the number of learned parameterization increases.

hands and manipulated objects from images, paving to way towards single-image reconstruction of several objects and their interaction. Finally, Mescheder et al. (2019); Mildenhall et al. (2020); Park et al. (2019a); Wang et al. (2018b) also use neural networks to model piece-wise linear approximations of continuous 3D representations, but model volumetric representations instead of surfaces.

More results We provide some additional results in Annexe A, including :

- 3D autoencoding experiments of human shapes.
- Super Resolution qualitative results on 3D chairs

- Detailed quantitative results, per category, for SVR and autoencoding experiments on ShapeNet
- More qualitative results on SVR and autoencoding
- Experiments with regularization
- Shape Correspondence qualitative results.

We also provide interactive results online:

- A 3D web server to compare AtlasNet with 3D-R2N2 Choy et al. (2016) and PointSetGen Fan et al. (2017), on SVR and autoencoding, showing 10 qualitative example for each of the 13 ShapeNet categories.
- Videos of 3D shape interpolation to complement the results of Figure 3.7a.

3.6 Conclusion

We have introduced an approach to generate parametric surface elements for 3D shapes. We have shown its benefits for 3D shape and single-view reconstruction, out-performing existing baselines. In addition, we have shown its promises for shape interpolation, finding shape correspondences, and mesh parameterization. Our approach opens up applications in generation and synthesis of meshes for 3D shapes, similar to still image generation Isola et al. (2017); Zhu et al. (2017).

The next direction we choose to explore more was the potential of AtlasNet to provide meaningful correspondences between shape. Indeed, as can be seen in Figure 3.7b, we can infer correspondences between two scans simply by reconstructing them from deformations of the same template surface. Those correspondences have the advantage of being obtained without any supervision at this point and are dense. In the next chapter, we are going to build on this and propose a method for dense correspondences that outperforms the state of the art by 15%.

Chapter 4

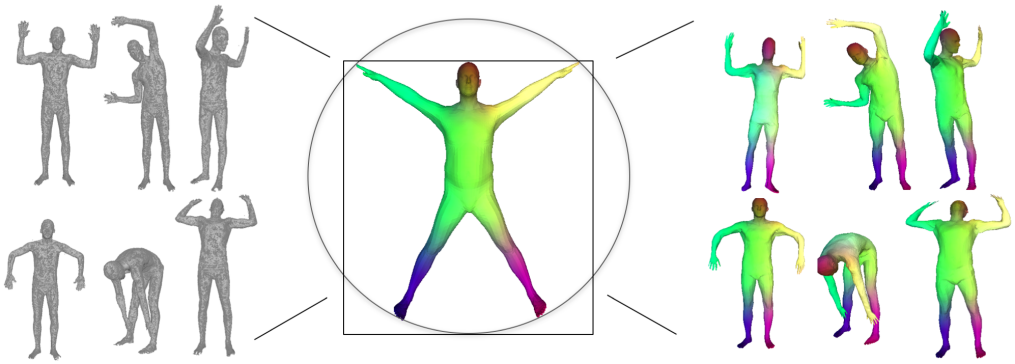
3D-CODED : 3D Correspondences by Deep Deformation

Abstract

In this Chapter, we show how the idea to learn surface deformations can be build extended to provide state-of-the-art shape matching results. We introduce *Shape Deformation Networks* which jointly encode 3D shapes and correspondences. This is achieved by factoring the surface representation into (i) a template, that parameterizes the surface, and (ii) a learnt global feature vector that parameterizes the transformation of the template into the input surface. By predicting this feature for a new shape, we implicitly predict correspondences between this shape and the template. We show that these correspondences can be improved by an additional step which optimize the shape feature by minimizing the Chamfer distance between the input and transformed template. We demonstrate that our simple approach improves on state-of-the-art results on the difficult FAUST challenge. We show, on the TOSCA dataset, that our method is robust to many types of perturbations, and generalizes to non-human shapes. This robustness allows it to perform well on real unclean, meshes from the the SCAPE dataset our outperform all other approaches in the SHREC 2019 shape matching challenge.

The work presented in this chapter was initially presented in:

"3D-CODED : 3D Correspondences by Deep Deformation", Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry, In *Proceedings of the European Conference on Computer Vision (ECCV 2018)*.



(a) Input Shape **(b) Template** **(c) Deformed template**

Figure 4.1 Our approach predicts shape correspondences by learning a consistent mesh parameterization with a shared template. Colors show correspondences.

4.1 Introduction

There is a growing demand for techniques that make use of the large amount of 3D content generated by modern sensor technology. A particular task that would be useful to analyze this data is to establish reliable 3D shape correspondences between scans from raw sensor data or between scans and a template 3D shape. This process is challenging due to low sensor resolution and high sensor noise, especially for articulated shapes, such as humans or animals, that exhibit significant non-rigid deformations and shape variations.

Traditional approaches to estimating shape correspondences for articulated objects typically rely on intrinsic surface analysis either optimizing for an isometric map or leveraging intrinsic point descriptors Sun et al. (2009a). To improve correspondence quality, these methods have been extended to take advantage of category-specific data priors Boscaini et al. (2016b). Effective human-specific templates and registration techniques have been developed over the last decade Zuffi and Black. (2015), but these methods require significant effort and domain-specific knowledge to design the parametric deformable template, create an objective function that ensures alignment of salient regions and is not prone to being stuck in local minima, and develop an optimization strategy that effectively combines a global search for a good heuristic initialization and a local refinement procedure.

In this chapter, we propose *Shape Deformation Networks*, a comprehensive, all-in-one solution to template-driven shape matching. A Shape Deformation Network learns to deform a template shape to align with an input observed shape. Given two input shapes, we align the template to both inputs and obtain the final map between the inputs by reading off the correspondences from the template.

We train our Shape Deformation Network as part of an encoder-decoder architecture, which jointly learns an encoder network that takes a target shape as input and generates a global feature representation, and a decoder Shape Deformation Network that takes as input the global feature and deform the template into the target shape. At test time, we improve our template-input shape alignment by optimizing locally the Chamfer distance between target and generated shape over the global feature representation which is passed in as input to the Shape Deformation Network. Critical to the success of our Shape Deformation Network is the ability to learn to deform a template shape to targets with varied appearances and articulation. We achieve this ability by training our network on a very large corpus of shapes.

In contrast to previous work Zuffi and Black. (2015), our method does not require a manually designed deformable template; the deformation parameters and degrees of freedom are implicitly learned by the encoder. Furthermore, while our network can take advantage of known correspondences between the template and the example shapes, which are typically

available when they have been generated using some parametric model Bogo et al. (2014); Varol et al. (2017), we show it can also be trained without correspondence supervision. This ability allows the network to learn from a large collection of shapes lacking explicit correspondences.

We demonstrate that with sufficient training data this simple approach achieves state-of-the-art results and outperforms techniques that require complex multi-term objective functions instead of the simple reconstruction loss used by our method.

4.2 Method

Our goal is, given a reference shape $\mathcal{S}_r$ and a target shape $\mathcal{S}_t$, to return a set of point correspondences $\mathcal{C}$ between the shapes. We do so using two key ideas. First, we learn to predict a transformation between the shapes instead of directly learning the correspondences. This transformation, from 3D to 3D can indeed be represented by a neural network more easily than the association between variable and large number of points. The second idea is to learn transformations only from one template $\mathcal{A}$ to any shape. Indeed, the large variety of possible poses of humans makes considering all pairs of possible poses intractable during training. We instead decouple the correspondence problem into finding two sets of correspondences to a common template shape. We can then form our final correspondences between the input shapes via indexing through the template shape. An added benefit is during training we simply need to vary the pose for a single shape and use the known correspondences to the template shape as the supervisory signal.

Our approach has three main steps which are visualized figure 5.2. First, a feed-forward pass through our encoder network generates an initial global shape descriptor (Section 4.2.1). Second, we use gradient descent through our decoder Shape Deformation Network to refine this shape descriptor to improve the reconstruction quality (Section 4.2.2). We can then use the template to match points between any two input shapes (Section 4.2.3).

4.2.1 Learning 3D shape reconstruction by template deformation

To put an input shape $\mathcal{S}$ in correspondence with a template $\mathcal{A}$, our first goal is to design a neural network that will take $\mathcal{S}$ as input and predict transformation parameters. We do so by training an encoder-decoder architecture. The encoder $\mathcal{E}_\phi$ defined by its parameters ϕ takes as input 3D points, and is a simplified version of the network presented in Qi et al. (2017a). It applies to each input 3D point coordinate a multi-layer perceptron with hidden feature size of 64, 128 and 1024, then maxpooling over the resulting features over all points followed by a linear layer, leading to feature of size 1024 $\mathcal{E}_\phi(\mathcal{S})$. This feature, together with the 3D coordinates of a

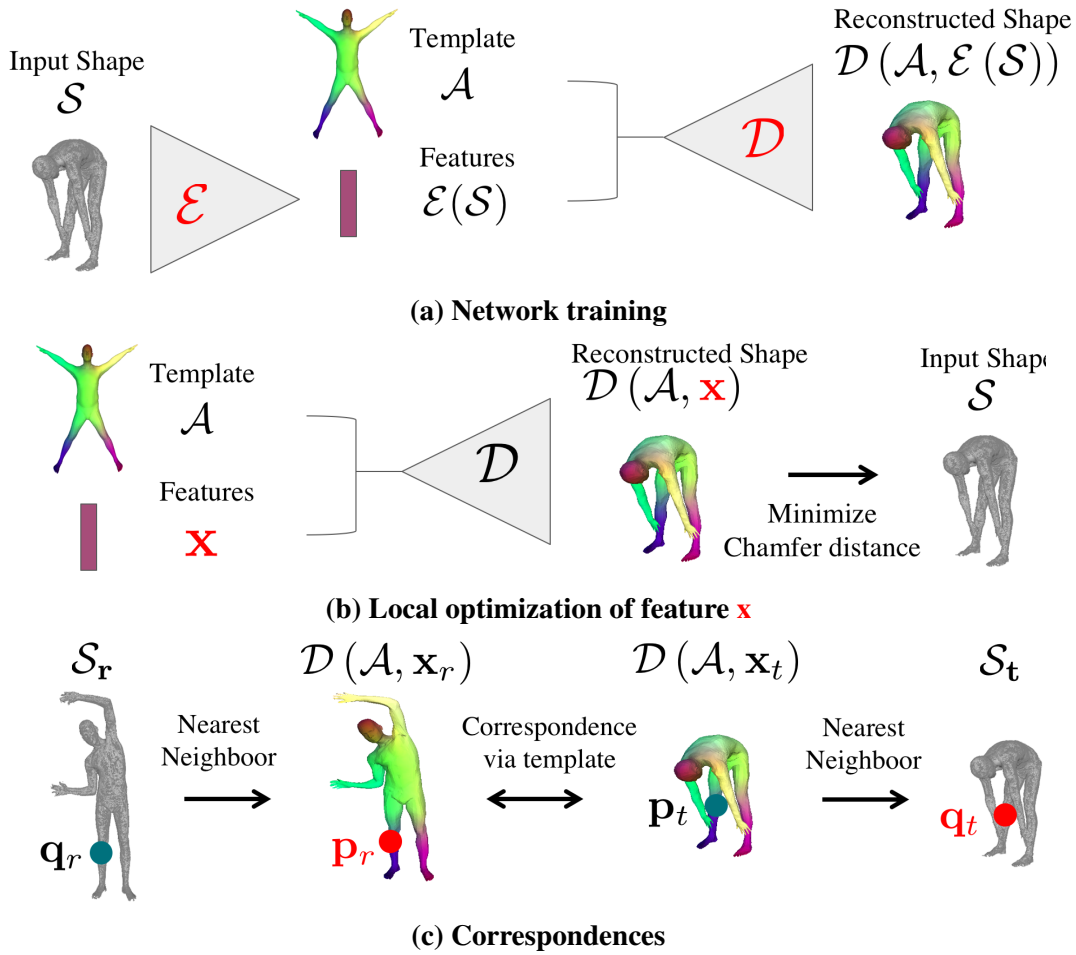


Figure 4.2 Method overview. (a) A feed-forward pass in our autoencoder encodes input point cloud $\mathcal{S}$ to latent code $\mathcal{E}(\mathcal{S})$ and reconstruct $\mathcal{S}$ using $\mathcal{E}(\mathcal{S})$ to deform the template $\mathcal{A}$. (b) We refine the reconstruction $\mathcal{D}(\mathcal{A}, \mathcal{E}(\mathcal{S}))$ by performing a regression step over the latent variable $\mathbf{x}$, minimizing the Chamfer distance between $\mathcal{D}(\mathcal{A}, \mathbf{x})$ and $\mathcal{S}$. (c) Finally, given two point clouds $\mathcal{S}_\mathbf{r}$ and $\mathcal{S}_\mathbf{t}$, to match a point $\mathbf{q}_r$ on $\mathcal{S}_\mathbf{r}$ to a point $\mathbf{q}_t$ on $\mathcal{S}_\mathbf{t}$, we look for the nearest neighbor $\mathbf{p}_r$ of $\mathbf{q}_r$ in $\mathcal{D}(\mathcal{A}, \mathbf{x}_r)$, which is by design in correspondence with $\mathbf{p}_t$; and look for the nearest neighbor $\mathbf{q}_t$ of $\mathbf{p}_t$ on $\mathcal{S}_\mathbf{t}$. Red indicates what is being optimised.

point on the template $\mathbf{p} \in \mathcal{A}$, are taken as input to the decoder $\mathcal{D}_\theta$ with parameters θ, which is trained to predict the position $\mathbf{q}$ of the corresponding point in the input shape. This decoder Shape Deformation Network is a multi-layer perceptron with hidden layers of size 1024, 512, 254 and 128, followed by a hyperbolic tangent. This architecture maps any points from the template domain to the reconstructed surface. By sampling the template more or less densely, we can generate an arbitrary number of output points by sequentially applying the decoder over sampled template points.

This encoder-decoder architecture is trained end-to-end. We assume that we are given as input a training set of N shapes $\left\{\mathcal{S}^{(i)}\right\}_{i=1}^{N}$ with each shape having a set of P vertices $\left\{\mathbf{q}_j\right\}_{j=1}^{P}$. We consider two training scenarios: one where the correspondences between the template and the training shapes are known (supervised case) and one where they are unknown (unsupervised case). Supervision is typically available if the training shapes are generated by deforming a parametrized template, but real object scans are typically obtained without correspondences.

4.2.1.1 Supervised loss.

In the supervised case, we assume that for each point $\mathbf{q}_j$ on a training shape we know the correspondence $\mathbf{p}_j \leftrightarrow \mathbf{q}_j$ to a point $\mathbf{p}_j \in \mathcal{A}$ on the template $\mathcal{A}$. Given these training correspondences, we learn the encoder $\mathcal{E}_\phi$ and decoder $\mathcal{D}_\theta$ by simply optimizing the following reconstruction losses,

$$\mathcal{L}^{\text{sup}}(\theta,\phi) = \sum_{i=1}^{N}\sum_{j=1}^{P} |\mathcal{D}_\theta\left(\mathbf{p}_j; \mathcal{E}_\phi\left(\mathcal{S}^{(i)}\right)\right) - \mathbf{q}_j^{(i)}|^2 \tag{4.1}$$

where the sums are over all P vertices of all N example shapes.

4.2.1.2 Unsupervised loss.

In the case where correspondences between the exemplar shapes and the template are not available, we also optimize the reconstructions, but also regularize the deformations toward isometries. For reconstruction, we use the Chamfer distance $\mathcal{L}^{\text{CD}}$ between the inputs $\mathcal{S}_i$ and reconstructed point clouds $\mathcal{D}_\theta\left(\mathcal{A}; \mathcal{E}_\phi\left(\mathcal{S}^{(i)}\right)\right)$. For regularization, we use two different terms. The first term $\mathcal{L}^{\text{Lap}}$ encourages the Laplacian operator defined on the template and the deformed template to be the same (which is the case for isometric deformations of the surface). The second term $\mathcal{L}^{\text{edges}}$ encourages the ratio between edges length in the template and its deformed version to be close to 1. More details on these different losses are given in B. The final loss we optimize is:

$$\mathcal{L}^{\text{unsup}} = \mathcal{L}^{\text{CD}} + \lambda_{Lap}\mathcal{L}^{\text{Lap}} + \lambda_{edges}\mathcal{L}^{\text{edges}} \tag{4.2}$$

where λ_{Lap} and λ_{edges} control the influence of regularizations against the data term $\mathcal{L}^{\text{CD}}$. They are both set to 5.10^{-3} in our experiments.

We optimize the loss using the Adam solver, with a learning rate of 10^{-3} for 25 epochs then 10^{-4} for 2 epochs, batches of 32 shapes, and 6890 points per shape.

One interesting aspect of our approach is that it learns jointly a parameterization of the input shapes via the decoder and to perdict the parameters $\mathcal{E}_\phi(\mathcal{S})$ for this parameterization via the encoder. However, the predicted parameters $\mathcal{E}_\phi(\mathcal{S})$ for an input shape $\mathcal{S}$ are not necessarily

optimal, because of the limited power of the encoder. Optimizing these parameters turns out to be important for the final results, and is the focus of the second step of our pipeline.

4.2.2 Optimizing shape reconstruction

We now assume that we are given a shape $\mathcal{S}$ as well as learned weights for the encoder $\mathcal{E}_\phi$ and decoder $\mathcal{D}_\theta$ networks. To find correspondences between the template shape and the input shape, we will use a nearest neighbor search to find correspondences between that input shape and its reconstruction. For this step to work, we need the reconstruction to be accurate. The reconstruction given by the parameters $\mathcal{E}_\phi(\mathcal{S})$ is only approximate and can be improved. Since we do not know correspondences between the input and the generated shape, we cannot minimize the loss given in equation (4.1), which requires correspondences. Instead, we minimize with respect to the global feature $\mathbf{x}$ the Chamfer distance between the reconstructed shape and the input:

$$\mathcal{L}^{\text{CD}}(\mathbf{x};\mathcal{S}) = \sum_{\mathbf{p}\in\mathcal{A}} \min_{\mathbf{q}\in\mathcal{S}} |\mathcal{D}_\theta(\mathbf{p};\mathbf{x}) - \mathbf{q}|^2 + \sum_{\mathbf{q}\in\mathcal{S}} \min_{\mathbf{p}\in\mathcal{A}} |\mathcal{D}_\theta(\mathbf{p};\mathbf{x}) - \mathbf{q}|^2 . \tag{4.3}$$

Starting from the parameters predicted by our first step $\mathbf{x} = \mathcal{E}_\phi(\mathcal{S})$, we optimize this loss using the Adam solver for 3,000 iterations with learning rate $5 * 10^{-4}$. Note that the good initialization given by our first step is key since Equation(4.3) corresponds to a highly non-convex problem, as shown in Figure 4.8.

4.2.3 Finding 3D shape correspondences

To recover correspondences between two 3D shapes $\mathcal{S}_r$ and $\mathcal{S}_t$, we first compute the parameters to deform the template to these shapes, $\mathbf{x}_r$ and $\mathbf{x}_t$, using the two steps outlined in section 4.2.1 and 4.2.2. Next, given a 3D point $\mathbf{q}_r$ on the reference shape $\mathcal{S}_r$, we first find the point $\mathbf{p}$ on the template $\mathcal{A}$ such that its transformation with parameters $\mathbf{x}_r$, $\mathcal{D}_\theta(\mathbf{p};\mathbf{x}_r)$ is closest to $\mathbf{q}_r$. Finally we find the 3D point $\mathbf{q}_t$ on the target shape $\mathcal{S}_t$ that is the closest to the transformation of $\mathbf{p}$ with parameters $\mathbf{x}_t$, $\mathcal{D}_\theta(\mathbf{p};\mathbf{x}_t)$. Our algorithm is summarized in Algorithm 2 and illustrated in Figure 5.2.

Algorithm 2: Algorithm for finding 3D shape correspondences

Input : Reference shape $\mathcal{S}_r$ and target shape $\mathcal{S}_t$
Output : Set of 3D point correspondences $\mathcal{C}$

#Regression steps over latent code to find best reconstruction of $\mathcal{S}_r$ and $\mathcal{S}_t$
$\mathbf{x}_r \leftarrow \arg\min_{\mathbf{x}} \mathcal{L}^{\text{CD}}(\mathbf{x}; \mathcal{S}_r)$ #detailed in equation (4.3)
$\mathbf{x}_t \leftarrow \arg\min_{\mathbf{x}} \mathcal{L}^{\text{CD}}(\mathbf{x}; \mathcal{S}_t)$ #detailed in equation (4.3)
$\mathcal{C} \leftarrow \varnothing$
Matching of $\mathbf{q}_r \in \mathcal{S}_r$ to $\mathbf{q}_t \in \mathcal{S}_t$
foreach $\mathbf{q}_r \in \mathcal{S}_r$ **do**
  $\mathbf{p} \leftarrow \arg\min_{\mathbf{p}' \in \mathcal{A}} |\mathcal{D}_\theta(\mathbf{p}'; \mathbf{x}_r) - \mathbf{q}_r|^2$
  $\mathbf{q}_t \leftarrow \arg\min_{\mathbf{q}' \in \mathcal{S}_t} |\mathcal{D}_\theta(\mathbf{p}; \mathbf{x}_t) - \mathbf{q}'|^2$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathbf{q}_r, \mathbf{q}_t)\}$
end
return $\mathcal{C}$

(a) SURREAL Varol et al. (2017) **(b)** Bent shapes **(c)** FAUST Bogo et al. (2014)

Figure 4.3 Examples of the different datasets used in the paper.

4.3 Results

4.3.1 Datasets

4.3.1.1 Synthetic training data.

To train our algorithm, we require a large set of shapes. We thus rely on synthetic data for training our model.

For human shapes, we use SMPL Bogo et al. (2014), a state-of-the-art generative model for synthetic humans. To obtain realistic human body shape and poses from the SMPL model, we sampled 2.10^5 parameters estimated in the SURREAL dataset Varol et al. (2017). One limitation of the SURREAL dataset is it does not include any humans bent over. Without adapted training data, our algorithm generalized poorly to these poses. To overcome this limitation, we generated an extension of the dataset. We first manually estimated 7 key-joint parameters (among 23 joints in the SMPL skeletons) to generate bent humans. We then sampled randomly the 7 parameters around these values, and used parameters from the SURREAL dataset for the other pose and body shape parameters. Note that not all meshes generated with

this strategy are realistic as shown in figure 4.3. They however allow us to better cover the space of possible poses, and we added $3 \cdot 10^4$ shapes generated with this method to our dataset. Our final dataset thus has $2.3 \cdot 10^5$ human meshes with a large variety of realistic poses and body shapes.

For animal shapes, we use the SMAL model, which provides the equivalent of SMPL for several animals Zuffi et al. (2017) . Recent papers estimate model parameters from images, but no large-scale parameter set is yet available. For training we thus generated models from SMAL with random parameters (drawn from a Gaussian distribution of *ad-hoc* variance 0.2). This approach works for the 5 categories available in SMAL. In SMALR, Zuffi et al. (2018) showed that the SMAL model could be generalized to other animals using only an image dataset as input, demonstrating it on 17 additional categories. Note that since the templates for two animals are in correspondences, our method can be used to get inter-category correspondences for animals. We qualitatively demonstrate this on hippopotamus/horses in the Annex B.

4.3.1.2 Testing data.

We evaluate our algorithm on the FAUST Bogo et al. (2014), TOSCA Bronstein et al. (2008), SHREC Dyke et al. (2019b) SCAPE Anguelov et al. (2005) datasets.

The FAUST dataset consists of 100 training and 200 testing scans of approximately 170,000 vertices. They may include noise and have holes, typically missing part of the feet. In this paper, we never used the training set, except for a single baseline experiment, and we focus on the test set. Two challenges are available, focusing on intra- and inter-subject correspondences. The error is the average Euclidean distance between the estimated projection and the ground-truth projection. We evaluated our method through the online server and are the best public results on the 'inter' challenge at the time of submission[1].

The SCAPE Anguelov et al. (2005) dataset has two sets of 71 meshes : the first set consists of real scans with holes and occlusions and the second set are registered meshes aligned to the first set. The poses are different from both our training dataset and FAUST.

The SHREC Dyke et al. (2019b) dataset has 19 pairs of models to be matched; the pairings are between a thin clothed mannequin and a larger mannequin, ensuring significant non-isometry.

TOSCA is a dataset produced by deforming 3 template meshes (human, dog, and horse). Each mesh is deformed into multiple poses, and might have various additional perturbations such as random holes in the surface, local and global scale variations, noise in vertex positions, varying sampling density, and changes in topology.

[1] http://faust.is.tue.mpg.de/challenge/Inter-subject_challenge

4.3.1.3 Shape normalization.

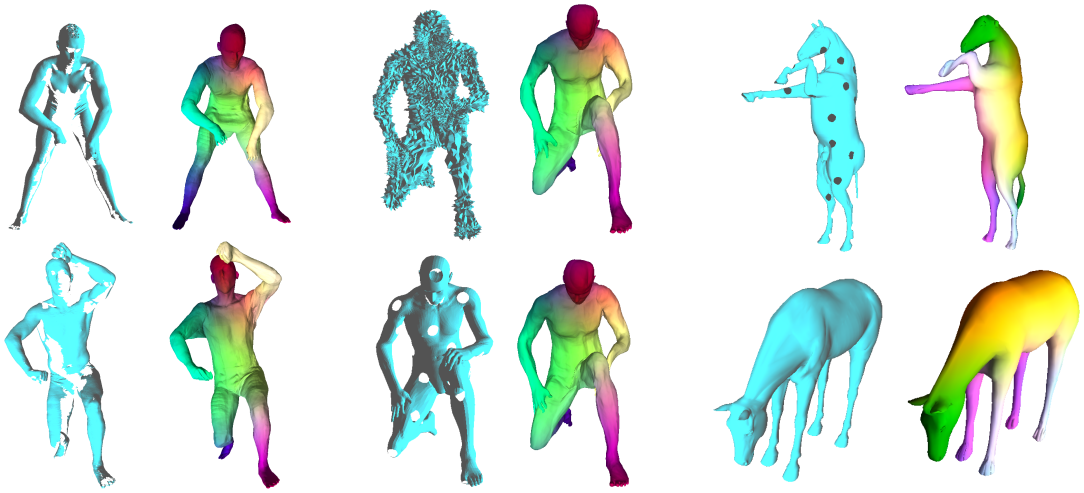
To be processed and reconstructed by our network, the training and testing shapes must be normalized in a similar way. Since the vertical direction is usually known, we used synthetic shapes with approximately the same vertical axis. We also kept a fixed orientation around this vertical axis, and at test time selected the one out of 50 different orientations which leads to the smaller reconstruction error in term of Chamfer distance. Finally, we centered all meshes according to the center of their bounding box and, for the training data only, added a random translation in each direction sampled uniformly between -3cm and 3cm to increase robustness.

4.3.2 Experiments

In this part, we analyze the key components of our pipeline.

4.3.2.1 Results on FAUST.

The method presented above leads to the best results to date on the FAUST-inter dataset: 2.878 cm : **an improvement of 8% over state of the art**, 3.12cm for Zuffi and Black. (2015) and 4.82cm for Litany et al. (2017). Although it cannot take advantage of the fact that two meshes represent the same person, our method is also the second best performing (average error of 1.99 cm) on FAUST-intra challenge.



(a) SCAPE Anguelov et al. (2005) **(b)** TOSCA Bronstein et al. (2008) **(c)** TOSCA animals Bronstein et al. (2008)

Figure 4.4 Other datasets. Left images show the input, right images the reconstruction with colors showing correspondences. Our method works with real incomplete scans **(a)**, strong synthetic perturbations **(b)**, and on non-human shapes **(c)**.

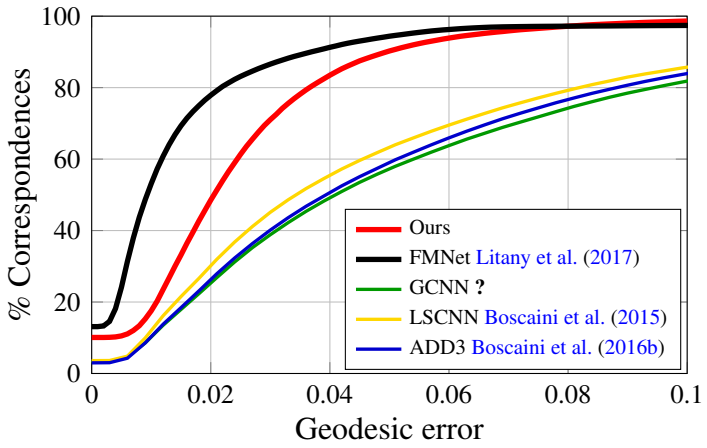


Figure 4.5 Comparison with learning-based shape matching approaches on the SCAPE dataset. Our method is trained on synthetic data, FMNet was trained on FAUST data, and all other methods on SCAPE. We outperform all methods except FMNet even though our method was trained on a different dataset.

4.3.2.2 Results on SCAPE : real and partial data.

The SCAPE dataset provides meshes aligned to real scans and includes poses different from our training dataset. When applying a network trained directly on our SMPL data, we obtain satisfying performance, namely 3.14cm average Euclidean error. Quantitative comparison of correspondence quality in terms of geodesic error are given in Fig 4.5. We outperform all methods except for Deep Functional Maps Litany et al. (2017). SCAPE also allows evaluation on real partial scans. Quantitatively, the error on these partial meshes is 4.04cm, similar to the performance on the full meshes. Qualitative results are shown in Fig 4.4a.

4.3.2.3 Results on SHREC and TOSCA : robustness to perturbations.

The TOSCA dataset provides several versions of the same synthetic mesh with different perturbations. We found that our method, still trained only on SMPL or SMAL data, is robust to all perturbations (isometry, noise, shotnoise, holes, micro-holes, topology changes, and sampling), except scale, which can be trivially fixed by normalizing all meshes to have consistent surface area. Examples of representative qualitative results are shown Fig 4.4b and quantitative results are reported in Annexe B.

The SHREC workshop challenge provides several mannequins to be matched in different poses and with different orientations. Our method, trained only on SMPL data, outperforms all other approaches compared in the workshop Bouaziz and Pauly (2013); Dyke et al. (2019a); Li et al. (2019); R3DS (2018); Sahillioğlu (2018); Vestner et al. (2017). This goes to show

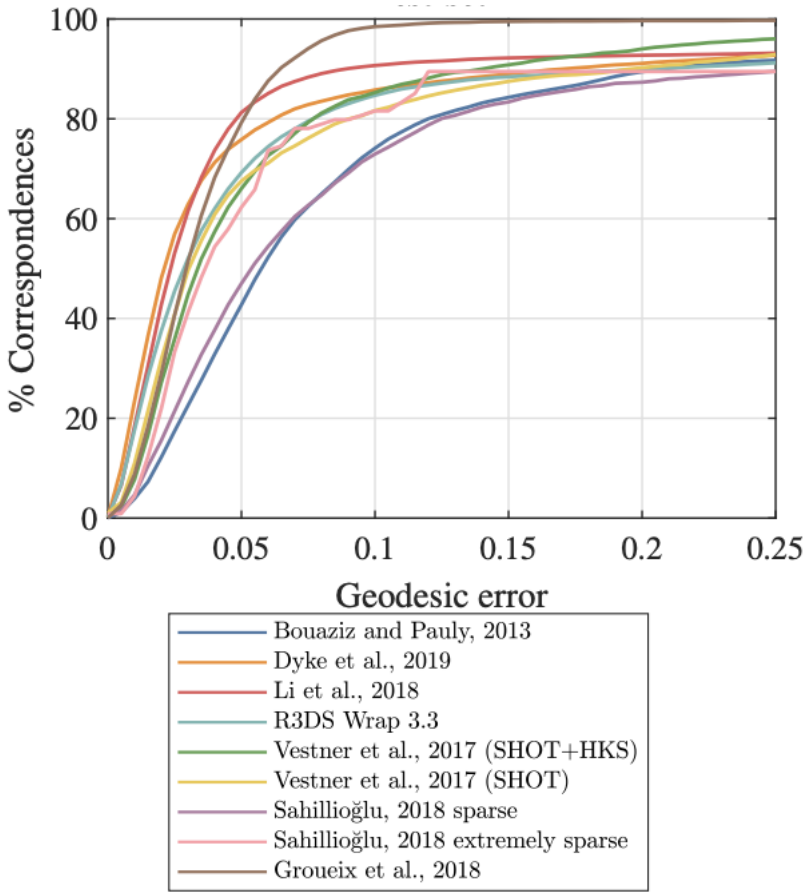


Figure 4.6 Figure from SHREC workshop Dyke et al. (2019b) summarizing a quantitative comparison of 3D-CODED with learning-based shape matching approaches. Our method is trained on SMPL data, and is the fastest to converge to 100%.

that 3D-CODED can generalised to other datasets. The quantitative comparison on SHREC is reported in Figure 4.6, while Figure illustrates the generalization capabilities of 3D-CODED on SHREC mannequins and a monstrous example provided by Zhongshi Jiang.

We thank Zhongshi Jiang who contributed a qualitative example of a monstrous shape that further highlight the generalization capabilities of the method.

4.3.2.4 Reconstruction optimization.

Because the nearest neighbors used in the matching step are sensitive to small errors in alignment, the second step of our pipeline which finds the optimal features for reconstruction, is crucial to obtain high quality results. This optimization however converges to a good optimum only if it is initialized with a reasonable reconstruction, as visualized in Figure 4.8. Since we optimize using Chamfer distance, and not correspondences, we also rely on the fact that the

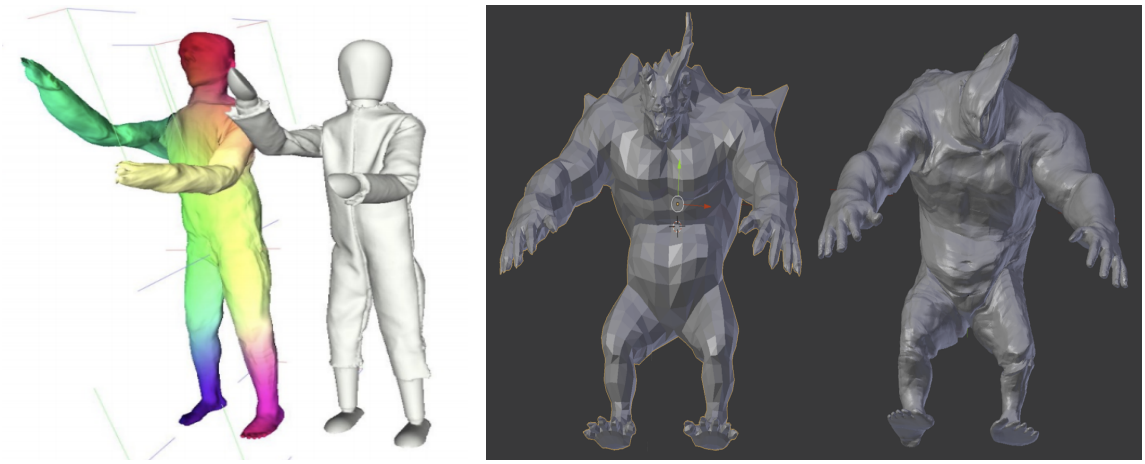


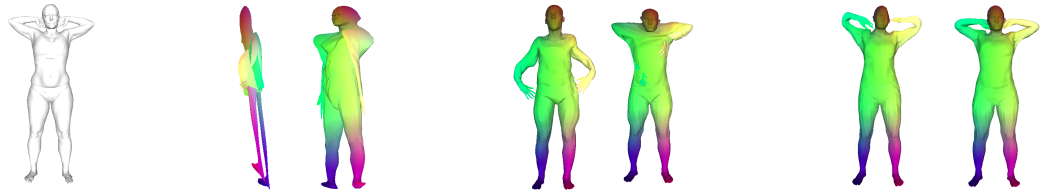
Figure 4.7 Two qualitative examples showing that our method is able to generalize beyond the SMPL data it was trained on. **Left.** Figure from SHREC workshop Dyke et al. (2019b) showing a mannequin reconstruction. **Right.** We thank Zhongshi Jiang who contributed this qualitative example of a reconstruction of a monstrous shape that further highlights the generalization capabilities of the method.

network was trained to generate humans in correspondence and we expect the optimized shape to still be meaningful.

Table 4.1 reports the associated quantitative results on FAUST-inter. We can see that: (i) optimizing the latent feature to minimize the Chamfer distance between input and output provides a strong boost; (ii) using a better (more uniform) sampling of the shapes when training our network provided a better initialization; (iii) using a high resolution sampling of the template (∼200k vertices) for the nearest-neighbor step provide an additional small boost in performance.

Method	Faust error (cm)
Without regression	6.29
With regression	3.255
With regression + Regular Sampling	3.048
With regression + Regular Sampling + High-Res template	**2.641**

Table 4.1 Importance of the reconstruction optimization step. Optimizing the latent feature is key to our results. Regular point sampling for training and high resolution for the nearest neighbor step provide an additional boost.



(a) Input **(b)** Random init. **(c)** Incorrect init. **(d)** Valid init.

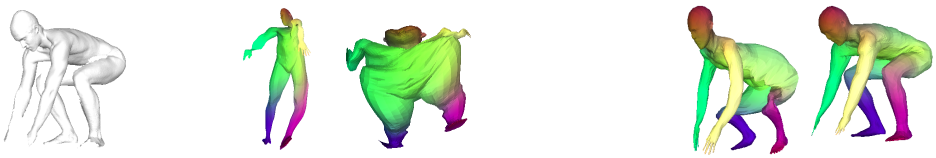
Figure 4.8 Reconstruction optimization. The quality of the initialization (i.e. the first step of our algorithm) is crucial for the deformation optimization. For a given target shape (a) and for different initializations (left of (b), (c) and (d)) the figure shows the results of the optimization. If the initialization is random (b) or incorrect (c), the optimization converges to bad local minima. With a reasonable initialization (d) it converges to a shape very close to the target ((d), right).

4.3.2.5 Necessary amount of training data.

Training on a large and representative dataset is also crucial for our method. To analyze the effect of training data, we ran our method without re-sampling FAUST points regularly and with a low resolution template for different training sets: FAUST training set, 2×10^5 SURREAL shapes, and 2.3×10^5, 10^4 and 10^3 shapes from our augmented dataset. The quantitative results are reported Table 4.2 and qualitative results can be seen in Figure 4.9. The FAUST training set only include 10 different poses and is too small to train our network to generalize. Training on many synthetic shapes from the SURREAL dataset Varol et al. (2017) helps overcome this generalization problem. However, if the synthetic dataset does not include any pose close to test poses (such as bent-over humans), the method will fail on these poses (4 test pairs of shapes out of 40). Augmenting the dataset as described in section 4.3.1 overcomes this limitation. As expected the performance decreases with the number of training shapes, respectively to 5.76cm and 4.70cm average error on FAUST-inter.

training data	Faust error (cm)
FAUST training set	18.22
non-augmented synthetic dataset 2×10^5 shapes	5.63
augmented synthetic data, 10^3 shapes	5.76
augmented synthetic data, 10^4 shapes	4.70
augmented synthetic data, 2.3×10^5 shapes	**3.26**

Table 4.2 FAUST-inter results when training on different datasets. Adding synthetic data reduce the error by a factor of 3, showing its importance. The difference in performance between the basic synthetic dataset and its augmented version is mostly due to failure on specific poses, as in Figure 4.3 .



(a) Input **(b)** FAUST training data **(c)** Augm. synth. training data

Figure 4.9 Importance of the training data. For a given target shape (a) reconstructed shapes when the network is trained on FAUST training set (b) and on our augmented synthetic training set (c), before (left) and after (right) the optimization step.

4.3.2.6 Unsupervised correspondences.

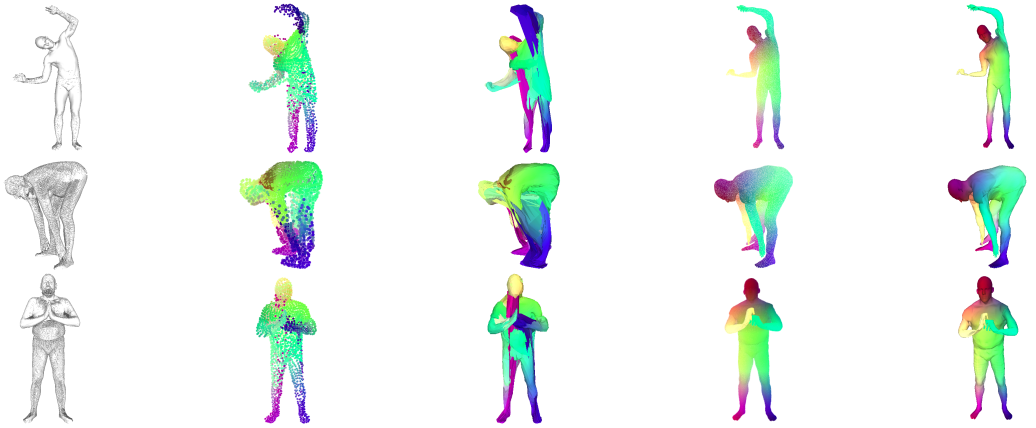
We investigate whether our method could be trained without correspondence supervision. We started by simply using the reconstruction loss described in Equation (4.3). One could indeed expect that an optimal way to deform the template into training shapes would respect correspondences. However, we found that the resulting network did not respect correspondences between the template and the input shape, as visualized figure 4.10. However, these results improve with adequate regularization such as the one presented in Equation (4.2), encouraging regularity of the mapping between the template and the reconstruction. We trained such a network with the same training data as in the supervised case but **without any correspondence supervision** and obtained a 4.88cm of error on the FAUST-inter data, i.e. similar to Deep Functional Map Litany et al. (2017) which had an error of 4.83 cm. This demonstrates that our method can be efficient even without correspondence supervision.

4.3.2.7 Rotation invariance

We handled rotation invariance by rotating the shape and selecting the orientation for which the reconstruction is optimal. As an alternative, we tried to learn a network directly invariant to rotations around the vertical axis. It turned out the performances were slightly worse on FAUST-inter (3.10cm), but still better than the state of the art. We believe this is due to the limited capacity of the network and should be tried with a larger network. However, interestingly,

Loss	Faust error (cm)
Chamfer distance, eq. 4.3 (unsupervised)	8.727
Chamfer distance + Regularization, eq. 4.2 (unsupervised)	4.835
Correspondences, eq. 4.1 (supervised)	**2.641**

Table 4.3 Results with and without supervised correspondences. Adding regularization helps the network find a better local minimum in terms of correspondences.



(a) Input (FAUST) **(b)** P.C. after optim. **(c)** Mesh after optim. **(d)** P.C. after optim+regul **(e)** Mesh after optim+regul

Figure 4.10 Unsupervised correspondences. We visualize for different inputs (a), the point clouds (P.C.) predicted by our approach (b,d) and the corresponding meshes (c,e). Note that without regularization, because of the strong distortion, the meshes appear to barely match to the input, while the point clouds are reasonable. On the other hand surface regularization creates reasonable meshes.

this rotation invariant network seems to have increased robustness and provided slightly better results on SCAPE.

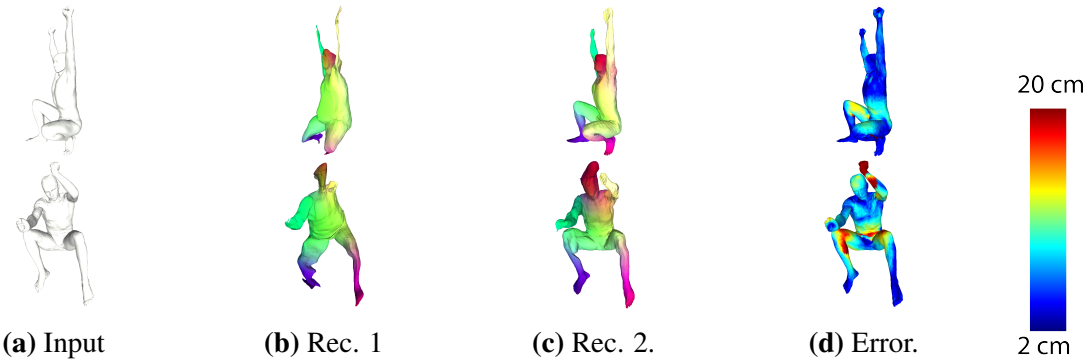
4.3.2.8 Failure cases

Figure 4.11 shows the two main sources of error our algorithm faces:

- Nearest-neighbor step in overlapping regions failure: a point is matched with the closest point in Euclidean distance but the match is very far in geodesic distance. This could be addressed by adding some regularity in the matches found by the nearest neighbor step. We leave this to future work.
- Failures in reconstruction: in such cases, the initial guess of the autoencoder is just too far away from the input, and the regression step fails.

More results We provide some additional results in Annexe B, including:

- Experiments with a different choice of template.
- Quantitative results against perturbations on TOSCA.
- Cross-category correspondences on animals.



(a) Input **(b)** Rec. 1 **(c)** Rec. 2. **(d)** Error.

Figure 4.11 Error visualization Given the input mesh (a), our autoencoder makes an initial reconstruction (b), optimized by a regression step (c). The average in centimeters over each vertex of (a), of the Euclidean distance between its projection and the ground truth, is reported (d). Red vertices have an error higher than 20cm, blue ones lower than 2cm. The largest error are observed in places where the Euclidean distance is small, while the geodesic distance is high, such as touching skin (zoom in on the leg). In such region, the nearest neighbors step is match a vertex in mesh A in a distant (in terms of geodesic distance) vertex in mesh A's reconstruction. High error can also come from a bad reconstruction, such as the head of the second example.

- Details on the regularization of the unsupervised loss.
- Experiments with asymmetric Chamfer distance

Note that are available online:

- Online benchmark result for the FAUST inter and intra challenges.
- The workshop paper detailing the comparison of 3D-CODED with other approaches on SHREC data Dyke et al. (2019b).

4.4 Conclusion

We have demonstrated an encoder-decoder deep network architecture that can generate human shape correspondences competitive with state-of-the-art approaches and that uses only simple reconstruction and correspondence losses. Our key insight is to factor the problem into an encoder network that produces a global shape descriptor, and a decoder Shape Deformation Network that uses this global descriptor to map points on a template back to the original geometry. A straightforward regression step uses gradient descent through the Shape Deformation Network to significantly improve the final correspondence quality.

A key element in our approach is the template to be deformed. For humans, we arbitrarily chose it to be a neutral human. In a subsequent work done by Theo Deprelle, we introduce two approaches to jointly learn the template surface deformations and learn the optimal template shape Deprelle et al. (2019). Learning the optimal template shapes yields better shape reconstructions and thus improves performances for shape matching.

In Chapter 3 and Chapter 4, we have introduced a new way to represent 3D shapes using deformations encoded by neural networks. Using this new data representation, we made an important bridge between the matching problem and 3D generation, and advance the state-of-the-art in both tasks.

Putting shapes in correspondences through a common template is not possible for man-made shapes, such as chairs, and table because of topological variations. In the next chapter, we extend the correspondence method to arbitrary categories of object and address two challenges: it is not possible to define a common template for some complex object categories, and such categories are also costly and difficult to annotate.

Chapter 5

Unsupervised cycle-consistent deformation for shape matching

Abstract

In this chapter, we propose a self-supervised approach to deep surface deformation in the absence of annotated data for correspondences. In contrast to chapter 4, we do not assume the existence of a template nor that pairs of shapes differ by an a near-isometric deformations. Given a pair of shapes from any category, our algorithm directly predicts a parametric transformation from one shape to the other respecting correspondences. Our insight is to use cycle-consistency to define a notion of good correspondences in groups of objects and use it as a supervisory signal to train our network. We demonstrate the efficacy of our approach by using it to transfer segmentation across shapes. We show, on Shapenet, that our approach is competitive with comparable state-of-the-art methods when annotated training data is readily available, but outperforms them by a large margin in the few-shot segmentation scenario.

The work presented in this chapter was initially presented in:

"Unsupervised cycle-consistent deformation for shape matching.", Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry, In *Computer Graphics Forum (SGP 2019)*.

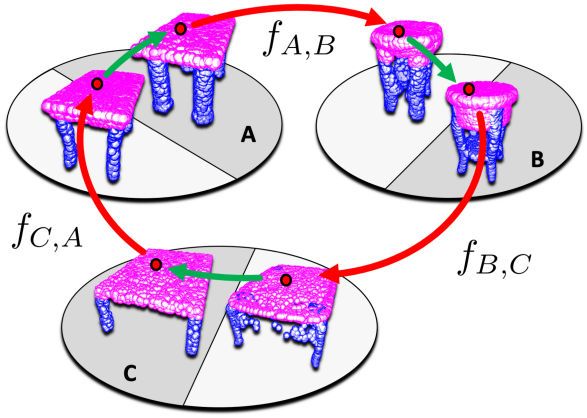


Figure 5.1 Shape deformation with cycle-consistency. Our approach takes a pair (A,B) of point-clouds as input and predicts a deformation of A into B. During training, a cycle-consistent loss on a shape triplet (A, B, C) allows the method to learn semantically consistent deformations $f_{A,B}$, $f_{B,C}$, $f_{C,A}$ without any priors. Red arrows represent the learned shape deformation function and green arrows indicate the projection of the deformed shape onto the nearest point on the surface of the target shape.

5.1 Introduction

Large collections of 3D models enable data-driven techniques for interactive geometry modeling, shape synthesis, image-based reconstruction, and shape completion Mitra et al. (2014). Many of these techniques require the collection to have additional surface annotations such as segmentation into functional Yi et al. (2016a) or geometric parts Li et al. (2018a). The notion of parts and their granularity can vary significantly across different tasks, so many novel applications require new types of annotations Mo et al. (2019a); Qi et al. (2017a); Wang et al. (2019b). Deep learning algorithms have recently achieved state-of-the-art in automatically predicting such surface annotations Qi et al. (2017a,b); Wang et al. (2018b). However, they typically require a significant number of training examples for every shape category, which limits their applicability, and bears significant start-up cost in introducing a new type of annotation. In this chapter, we propose a new deep learning approach which leverages large non-annotated object collections to perform few-shot segmentation.

We rely on the idea to use shape matching to transfer labels from similar examples. This approach has been shown to be robust in extreme "few-shot" learning scenarios Yi et al. (2016a) and can work robustly even in heterogeneous datasets as long as labeled models roughly span all the shape variations. The few-shots segmentation problem then amount to the fundamental problem of identifying correspondences between shapes. There is a vast amount of work on shape matching, which can be roughly separated in two trends: (i) classical optimization based approaches; (ii) recent approaches where correspondences are directly predicted by a neural network.

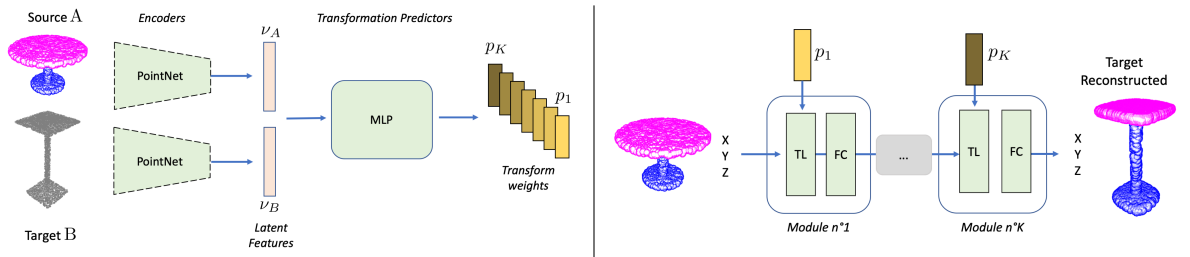
Traditional, optimization-based methods such as iterative closest point (ICP) algorithm, are fast and effective with good initial guesses and few degrees of freedom (e.g., a rigid motion) Rusinkiewicz and Levoy (2001). More flexible correspondence algorithms for dissimilar models usually require significantly more compute time to optimize for larger number of degrees of freedom Brown and Rusinkiewicz (2007); Chen and Koltun (2015); Kim et al. (2011). Since directly matching dissimilar shapes poses significant challenges, these methods often rely on joint analysis of the entire collection Kim et al. (2012), leveraging cycle consistency priors during optimization Huang and Guibas (2013); Nguyen et al. (2011). These joint correspondence estimation methods tend to be very compute heavy and as new models are added to the collection, the entire optimization needs to be repeated. We thus turned to deep learning-based approaches.

Indeed, with the recent advances in neural networks for geometry analysis, learning-based methods have been proposed to address the matching problem. We have proposed in Chapter 4 a deep approach to learn to deform a human body template to the target point cloud, even without

correspondence supervision. This approach is efficient, however it has to be trained specifically for each template, limiting this method to analysis of geometrically and topologically similar shape collections, such as human bodies. If such a template is not available, we showed with AtlasNet in Chapter 3, that one can pick a very generic shape (e.g., a sphere) and still obtain some notion of correspondences.

In this chapter, we propose a novel neural network architecture that learns to match shapes directly, without relying on a pre-defined template, by learning to predict deformations that aligns points on the source shape to points on the target. Note that the transformation can be much more complex than a rigid transformation, and that the space of meaningful transformation is defined implicitly by the (unlabelled) training data. We encode both source and target shapes and then predict the deformed position for every point on the source conditioned on these two codes, unlike prior work that use a fixed template common to all the shapes. We show that the results obtained can be greatly improved if the network is trained not only with a reconstruction loss, which encourages it to deform the source shape into the target shape, but also using a cycle consistency loss. Indeed a deformation which respects correspondences should be consistent between pairs of shapes *i.e.*, the deformation from A to B should be the inverse of the deformation from B to A . More generally, in larger cycles of shapes $[A_1, ..., A_i, .., A_N]$, global consistency is achieved if the composition of the N successive mappings from A_i to A_{i+1} is identity. This new consistency loss used during training can be seen as playing a role similar to the global consistency objective used in optimization-based approaches. Finally, our network is trained in an self-supervised manner using only shape reconstruction and cycle consistency losses.

We demonstrate the effectiveness of our approach for shape matching by propagating segmentations in a few-shot learning setting on the ShapeNet part dataset Yi et al. (2016a). We first show that in this extreme case with very few training examples, PointNet Qi et al. (2017a), a strongly supervised method, fails to generalize. Then, we propose several strategies for picking source shapes and propagate the signal from them, using our predicted correspondences. We demonstrate that even with a simple strategy, such as picking the source with smallest Chamfer distance, our method is better at transferring segmentations than other fast correspondence techniques such as ICP with rigid transformation and a learning-based method such as AtlasNet, that aligns sphere and plane templates.



(a) Parameter prediction network.

(b) Deformation network.

Figure 5.2 Shape Deformation approach. Our methods take as input a pair (source A, target B) of shapes and aims at predicting the deformation of A in B. In **(a)**, A and B are encoded with Pointnets Qi et al. (2017a) into a latent feature vector, from which an MLP predicts transformation parameters, used in **(b)** to deform A into B, by stacking Transformation Layers **(TL)** and Fully-Connected Layers **(FC)**.

5.2 Related Work

The related work on dense shape correspondences and 3D shape segmentation in discussed in Section 2.2. We discuss methods using cycle-consistency as a supervisory signal and discuss the specific case of few-shot segmentation.

Cycle-consistent correspondences Nguyen et al. (2011) propose to use cycle-consistency in a joint optimization to refine pairwise correspondences in a shape collection. Cycle-consistency was also use in the context of deep learning by Zhou et al. (2016) to train deep networks to predict correspondences between images of different instances of objects from the same category. In this work, views rendered from different viewpoints from a 3D model were used to avoid the trivial identity flow solution, but no correspondence between 3D shapes was predicted.

Few-shot mesh segmentation We demonstrate the value of our method for few-shot segmentation transfer. While many techniques have been developed for strongly supervised mesh segmentation Kalogerakis et al. (2017, 2010); Li et al. (2018a); Qi et al. (2017a,b); Wang et al. (2018b), they typically rely on many training examples and fail in a few-shot scenarios (see Table 5.1). In these cases, some framework propose to rely on propagating annotations from most similar annotated shapes via global or local shape matching Yi et al. (2016a). In fact, it is common for correspondence techniques to be evaluated and used for transferring various signals between shapes Azencot et al. (2017); Kim et al. (2011); Ovsjanikov et al. (2012).

5.3 Learning asymmetric cycle-consistent shape matching

We address the surface matching problem by training a model that takes as inputs a source shape, a target shape, and a point on the source shape and generates the corresponding point on the target shape. As pointed out in Chapter 3, a learnable model allows for efficient surface matching, which is in contrast to approaches requiring optimization over a collection of pairwise shape matches Nguyen et al. (2011).

We assume that shapes are represented as point sets sampled from the shapes' surface. Given point sets A and B, our goal is to learn a mapping function $f_{A,B}$ that takes a 3D point $\mathbf{p} \in A$ to its corresponding point $\mathbf{q} \in B$. If f is a function on points and A a set of points, we denote by $f(A)$ the set $\{f(\mathbf{p}), \forall \mathbf{p} \in A\}$.

First, similarly to our work on unsupervised template-based shape correspondence presented in Chapter 4, we use a Chamfer loss to minimize the distance between deformed source $f_{A,B}(A)$ and the target B. Unlike prior work, however, we do not assume that all of our shapes are derived from the same template and directly predict template-free correspondences between pairs of shapes.

Second, we seek to leverage the success of cycle consistency, which has been used in shape collection optimization Nguyen et al. (2011) and more recently in self-supervised learning **?**, during training of our learnable mapping function. Formally, for N shapes $X_1, \ldots, X_N$ that are assumed to be put into correspondence, we enforce that the learnable mapping function $f_{A,B}$ satisfies,

$$\forall \mathbf{p} \in X_1,\ f_{X_1,X_2} \circ \cdots \circ f_{X_{N-1},X_N} \circ f_{X_N,X_1}(\mathbf{p}) = \mathbf{p}. \tag{5.1}$$

We use cycle-consistency training losses for cycles of lengths two and three as it implies consistency for cycles of any length Nguyen et al. (2011). We visualize our cycle-consistency loss in Figure 5.1.

5.4 Approach

We describe our learnable mapping function $f_{A,B}$, implemented as a two-stage neural network, in Section 5.4.1, our training losses in Section 5.4.2, and application to segmentation in Section 5.4.3.

5.4.1 Architecture

The architecture of our shape transformation model from a source shape A to a target shape B is visualized in Figure 5.2 and can be separated into two parts: (a) a parameter prediction

network which outputs transformation parameters given the two shapes (Figure 5.2a); (b) a deformation network that transforms the first shape into the second one using the predicted parameters (Figure 5.2b). We now describe these two components.

To predict transformation parameters, A and B are first passed into two independent PointNet networks Qi et al. (2017a) leading to feature encodings v_A and v_B of size 512. The resulting concatenated descriptor $v_{AB} = [v_A, v_B]$ contains information about the pair (A, B). A multilayer perceptron (MLP) then predicts transformation parameters vectors $p_1, \cdots, p_K$ from this concatenated feature.

The deformation network (Figure 5.2b) takes a surface point in $\mathbb{R}^3$ and outputs the associated deformed point. The network is composed of K modules each with the same architecture. Let's call x_{k-1} the input of module k and x_k its output. The operation computed by this module is:

$$x_k = A_k\left(W_k\left(s_k \cdot x_{k-1} + b_k\right)\right), \tag{5.2}$$

where W_k is the matrix of parameters of a fully-connected layer in $\mathbb{R}^{64x64}$, "$\cdot$" refers to the Hadamard (term to term) product, A_k is the activation function for module k and $[s_k, b_k] = p_k$ are the transformation parameters, both in in $\mathbb{R}^{64}$, corresponding to a scale and a bias in each dimension. Note that this is similar to the architecture of the T-net modules in Jaderberg et al. (2015); Qi et al. (2017a), but using fewer predicted parameters. Also note that equation 5.2 is differentiable, which enables the two sub-networks to be trained jointly in an end-to-end fashion. In all of our experiments we used $K = 7$ modules, 64 dimensions for each intermediary feature and ReLU activations for all but the last layer, for which we used a hyperbolic tangent. We train for 500 epochs with Adam Kingma and Ba (2014) starting with a learning rate of 0.01 divided by 10 after 400 epochs.

5.4.2 Training Losses

We train our deformation by minimizing the sum over several components: a loss enforcing cycle consistency $L_{\mathbf{Cy}}$, Chamfer distance loss $L_{\mathbf{Ch}}$, and a self reconstruction loss $L_{\mathbf{SR}}$:

$$L_{\mathbf{total}} = L_{\mathbf{SR}} + L_{\mathbf{Ch}} + L_{\mathbf{Cy}}$$

We only use the self-reconstruction loss to stabilize the beginning of the training and disable it after 30 epochs to focus on cycle consistency and reconstruction losses. We train all parameters in our network by sampling triplets (A, B, C) of shapes which are needed by our 3-cycle consistency and enforcing all other losses on all the associated deformations. We first explain how we sampled these triplets, then detail the different terms of our loss.

5.4.2.1 Training shape sampling

For our cycle-consistency loss, we require a valid mapping across shape triplet (A, B, C). As different shape categories may have different topologies, we train category-specific networks. Furthermore, as there may be topological changes within a single category, for shape A, we randomly sample shapes B and C from the K nearest neighbors of A under chamfer distance. We take $K = 20$ and demonstrate in the ablation study the superiority of this approach over random sampling of shape triplets.

We apply data augmentation ψ on each sampled shape in this order : a random rotation around the Z axis of a random angle between $-40 and 40$, an anisotropic scaling of random scale between 0.75 and 1.25, a bounding box normalization, and a small random translation below 0.03.

5.4.2.2 Cycle-consistency loss

The cycle consistency loss is based on the intuition that a point deformed through any cycle of deformations should be mapped back to itself. One way to enforce consistency would be to compute composite functions, for two shapes X and Y minimizing $\|\mathbf{p} - f_{Y,X} \circ f_{X,Y}(\mathbf{p})\|$ for all $\mathbf{p}$ in X. However $f_{X,Y}(\mathbf{p})$ is typically not an element of Y, and computing $f_{Y,X} \circ f_{X,Y}(\mathbf{p})$ would thus require computing the deformations $f_{Y,X}$ of other points than the points of Y. To avoid this, we consider instead projections of the deformed shapes to the target shapes. More precisely, we define the shape projection operator π

$$\pi_X(\mathbf{p}) = \text{argmin}_{\mathbf{q} \in X} \|\mathbf{p} - \mathbf{q}\| \tag{5.3}$$

and enforce 2-cycle consistency between X and Y by minimizing

$$Cy_2(X,Y) = \frac{1}{|X|} \sum_{\mathbf{p} \in X} |\mathbf{p} - f_{Y,X} \circ \pi_Y \circ f_{X,Y}(\mathbf{p})|_2 \tag{5.4}$$

and cycle consistency for the (X,Y,Z) cycle by minimizing

$$Cy_3(X,Y,Z) = \frac{1}{|X|} \sum_{\mathbf{p} \in X} |\mathbf{p} - f_{Z,X} \circ \pi_Z \circ f_{Y,Z} \circ \pi_Y \circ f_{X,Y}(\mathbf{p})|_2 \tag{5.5}$$

Our full cycle-consistency loss $L_{\mathbf{Cy}}$ is simply defined by summing over possible all possible two and three cycles using a sampled triplet (A, B, C).

$$L_{\mathbf{Cy}} = \sum_{X,Y,Z \in \{A,B,C\} s.t. \{X,Y,Z\} = \{A,B,C\}} Cy_2(X,Y) + Cy_3(X,Y,Z) \tag{5.6}$$

Enforcing 2- and 3-cycle consistency implies consistency for any cycle Nguyen et al. (2011).

5.4.2.3 Reconstruction loss

As discussed in section 5.3, we want to enforce that every point in the target shape is well reconstructed, but not necessarily that any point in the source shape is mapped to the target shape, in case some part appear in the source and not the target. We thus used asymmetric Chamfer distance to quantify how well the network has generated the target shape. More precisely, given a pair of shapes (X,Y), the asymmetric chamfer $Ch(X,Y)$ computes the average distance between a point $\mathbf{q} \in Y$ and its nearest neighbor in X.

$$Ch(X,Y) = \frac{1}{|X|} \sum_{\mathbf{q} \in Y} \min_{\mathbf{p} \in X} \|p - q\|_2 . \tag{5.7}$$

Given a training triplet (A,B,C), we define the reconstruction loss by summing the asymmetric chamfer loss on all 6 possible (source, target) couples.

$$L_{\mathbf{Ch}} = \sum_{X,Y \in \{(A,B),(A,C),(B,C)\}} Ch(f_{X,Y}(X),Y) + Ch(f_{Y,X}(Y),X) \tag{5.8}$$

If segmentation is available for the training shapes, we can compute the distance in equation 5.7 on each segment independently, which would add supervision on the correspondences. We of course do not use such labels for our few-shot learning experiments, but show in Table 5.2 it can be used if available to slightly boost our results.

5.4.2.4 Self-reconstruction loss

We can fully supervise the deformation by manually deforming a shape with a known transformation. We found such a supervision was helpful to stabilize and speed up the beginning of our training. Concretely, we sampled deformations ψ similar to what we did for data augmentation (described above in 5.4.2.1) by composing (1) a rotation, (2) an anisotropic scaling, and (3) a rescaling to a centered bounding box. Given a transformation ψ, we compute the average distance between the two images of a point $\mathbf{p} \in A$ under ψ and the predicted mapping function $f_{A,\psi(A)}$.

$$SR(A,\psi) = \frac{1}{|A|} \sum_{\mathbf{p} \in A} \left\| f_{A,\psi(A)}(p) - \psi(p) \right\|_2 \tag{5.9}$$

Our corresponding self-reconstruction loss $L_{\mathbf{SR}}$ is the sum of this loss for each of the three point clouds in the triplet (A, B, C) with different random transformations.

$$L_{\mathbf{SR}} = SR(A, \psi) + SR(B, \psi') + SR(C, \psi'') \tag{5.10}$$

5.4.3 Application to segmentation

Learning a deformation between two shapes provides an intuitive method to transfer label information, such as a part segmentation, from a labeled shape to an unlabeled one. In this formulation, we assume we are given a (small) number of labeled shapes, and seek to label each point on an unlabeled test shape. This requires us to decide which of the labeled shapes we should use as the source to propagate labels to the target shapes.

Selection Criteria. Given a target T, We manually define 4 possible source selection criteria:

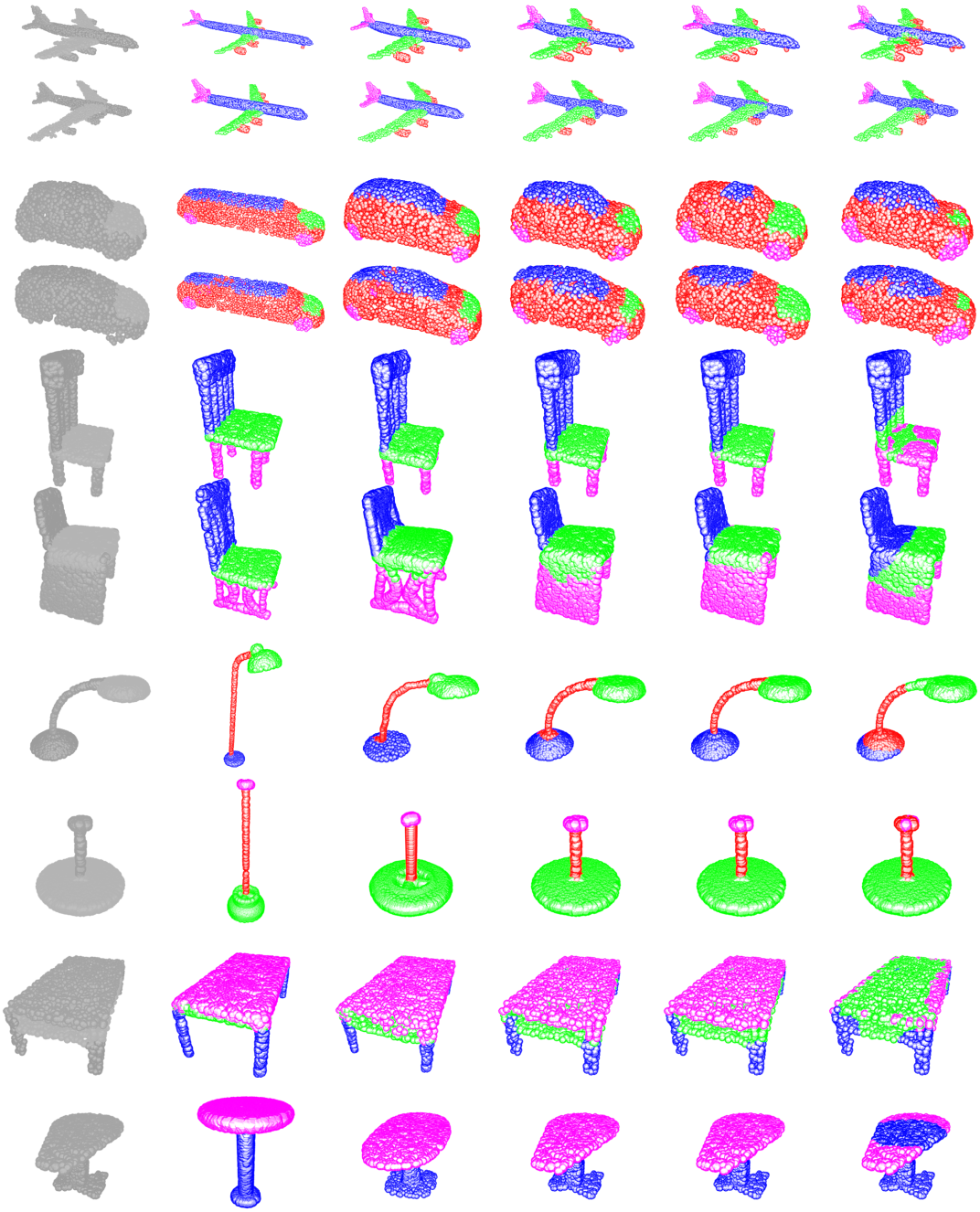
- **Nearest Neighbor**: The source shape S that minimizes the Chamfer distance between S and T is selected.
- **Deformation Distance**: The source shape S that minimizes the Chamfer distance between $f_{S,T}(S)$ and T is selected.
- **Cosine Distance**: The source shape S that minimizes the cosine distance distance between the PointNet encodings v_S and v_T is selected.
- **Cycle Consistency**: The source shape S that minimizes 2-cycle loss for the pair (S,T) is selected.

Having selected a pair (S,T), labels can be transferred directly with our approach.

Voting strategy. Instead of selecting a single source shape to get labels from, combining several voting shapes allows for better segmentation. We select the K-best sources, and make each source shape vote with equal weight for the label of each target point. We evaluate the benefits of this voting approach in Section 5.5.2.2.

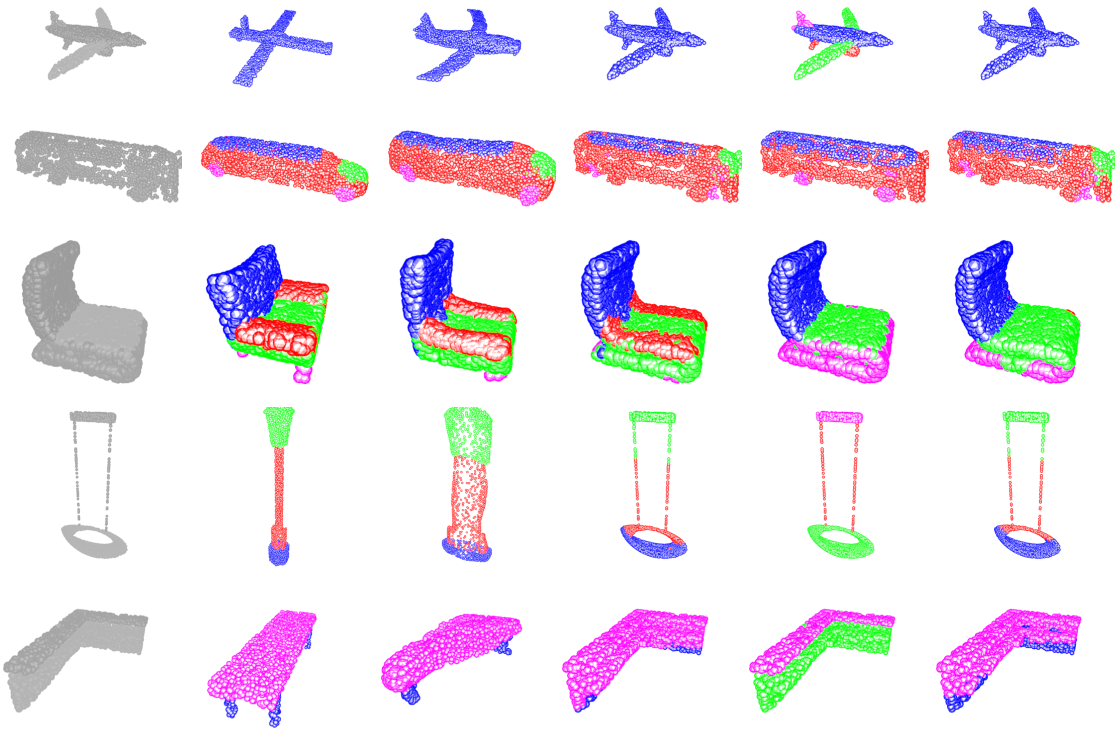
5.5 Results

In this section, we show qualitative and quantitative results on the tasks of few-shot and supervised semantic segmentation and compare against several baselines.



(a) Input shape (target) | **(b)** Retrieved shape (source) | **(c)** Deformed retrieved shape | **(d)** Labels transferred to input | **(e)** Ground truth labels | **(f)** Identity baseline segmentation

Figure 5.3 Qualitative results. For each input shape (a), we select the top nearest neighbor from 400 training examples with part segmentations using the cycle-consistency criterion (b). We apply our approach to deform the retrieved shape to align with the input shape (c). Given the deformed shape, we transfer the labels onto the input shape (d). For each category, we show the top results that maximize IoU with the ground truth (e). For comparison, we show the Identity baseline in (f). Notice how our method successfully transfers labels and improves over the baseline.



(a) Input shape (target) **(b)** Retrieved shape (source) **(c)** Deformed retrieved shape **(d)** Labels transferred to input **(e)** Ground truth labels **(f)** Identity baseline segmentation

Figure 5.4 Failures. Example failures include when a retrieved shape has inconsistent annotation (rows 1,2,5) and poor deformation due to different topology (rows 3,4).

Data and evaluation criteria. We evaluated our approach on the standard ShapeNet part dataset Yi et al. (2016a). We restricted ourselves to the 5 most populated categories, namely Airplane, Car, Chair, Lamp, and Table. Point clouds sampled on mesh objects are densely labeled for segmentation with one to five parts. We follow Qi et al. Qi et al. (2017a) and report the mean intersection over union (mIoU) between the predicted and ground truth segmentation across instances in a category.

Baselines. We compare our unsupervised approach against supervised and unsupervised approaches. We used PointNet as a supervised baseline. Our unsupervised baselines include a learned approach derived from Atlasnet and variants of iterative closest points (ICP) Besl et al. (1992); Zhang (1994). As presented in Chapter 3, AtlasNet is a template-based reconstruction method that predicts a transformation of the template matching the target shape. The learned deformations have been observed to be semantically consistent. To transfer segmentation labels

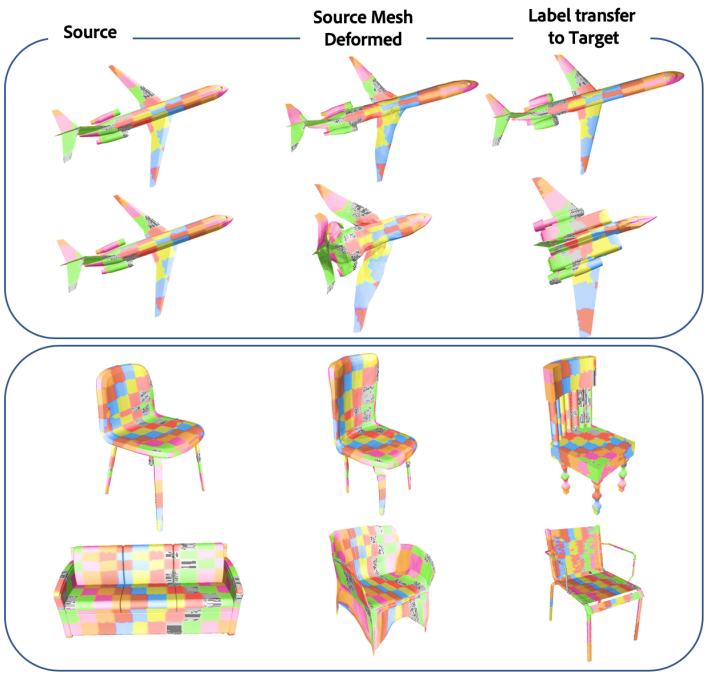


Figure 5.5 Mapping function quality. We apply a checkerboard colorization scheme on the source (**left**), and use our approach to deform (**middle**) the source shape to the target shape (**right**). The labels are transferred from the deformed shape to the target shape through nearest neighbors. For each category, we show a example of good reconstruction (**top**) and poor reconstruction (**bottom**). Notice the high quality of the mapping in both cases.

from a source to a target, we project the source labels on the source reconstruction through nearest neighbors, then on the template through dense correspondence between the template and the source reconstruction. Similarly, we transfer labels on the template to the target by dense correspondence and nearest neighbors. AtlasNet is trained on the same train/test splits as our approach. We consider two settings of AtlasNet – with 10 patches or 1 sphere as the template. Additionally, we use two standard shape alignment baselines. First, labels can be transferred from source to target through nearest neighbor matching, which we call the *Identity* baseline. An immediate refinement over this baseline is to apply ICP to align the source to the target, and then use nearest neighbors. We call the latter the *ICP* baseline.

5.5.1 Qualitative Results

Correspondences. In figure 5.5 we visualize in more detail the correspondences obtained with our approach. We visualize how each point on the source shape is deformed and transferred

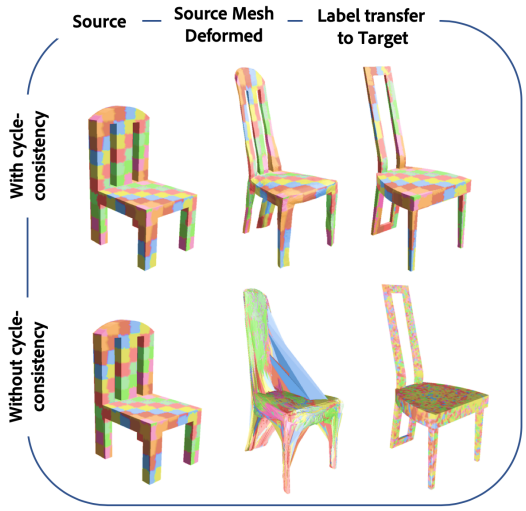


Figure 5.6 Cycle-consistency performance. We apply a checkerboard colorization scheme on the source (**left**), and use our approach with cycle-constistency (**top**) and without (**bottom**) to deform (**middle**) the source shape to the target shape (**right**). The labels are transferred from the deformed shape to the target shape through nearest neighbors.

to the target shape using a colored checkerboard. For each example, we show a successful deformation (top) and a failure case (bottom). Note how the checkerboard appears nicely deformed in the case of successful deformation, and still appears consistent on some parts in the failure cases.

Cycle-consistency. In figure 5.6 we compare the mappings learned by our approach with and without cycle-consistency loss. The Chamfer Distance is a point based loss with no control over the amount of distorsion. Notice in this case that the deformed source has large triangles. It indicates that the mapping learned by a Chamfer loss alone is not smooth, and can't be used in label tranfer. On the other hand, the cycle-consistency loss leads to a smooth and high quality mapping.

Segmentation transfer. When looking at the results, a first surprising observation is the high quality of the identity baseline (this is quantitatively confirmed in Table 5.2). Indeed, the different criteria tend to select shapes that are really close to the target. To focus on interesting examples, we selected in Figure 5.3 the pairs that maximize the performance improvement provided by our method compared to the identity baseline using the cycle-consistency-selection

10 shots	Selection Criterion	Airplane	Car	Chair	Lamp	Table
(a) Pointnet	-	14.0 ± 8.0	11.7 ± 10.4	21.1 ± 13.1	26.0 ± 13.2	43.5 ± 15.5
(b) Atlasnet Patch	Nearest Neighbors	62.6 ± 2.4	52.3 ± 9.1	72.1 ± 1.2	62.8 ± 2.2	61.6 ± 3.7
(c) Atlasnet Sphere	Nearest Neighbors	62.2 ± 2.2	52.9 ± 9.1	70.2 ± 1.2	59.3 ± 1.8	60.0 ± 5.1
(d) ICP	Nearest Neighbors	65.5 ± 3.1	61.3 ± 1.1	75.8 ± 1.2	64.8 ± 5.0	64.9 ± 3.9
(e) Ours	Nearest Neighbors	$\mathbf{67.1 \pm 2.9}$	$\mathbf{61.4 \pm 1.1}$	$\mathbf{78.9 \pm 1.1}$	$\mathbf{65.8 \pm 5.2}$	$\mathbf{66.1 \pm 4.5}$
(f) Ours	Cycle Consistency	67.9 ± 3.0	60.2 ± 3.4	81.8 ± 0.7	69.1 ± 5.4	68.8 ± 4.0
(g) Ours	Oracle	74.9 ± 3.0	68.6 ± 2.4	86.4 ± 0.6	80.3 ± 3.8	77.8 ± 2.1

Table 5.1 Few-shot segmentation:. We compare **(e, f)** our approach with **(a)** Pointnet Qi et al. (2017a), a supervised method, trained per category, **(b, c)** two unsupervised baselines based on Atlasnet and **(e)** ICP. We pre-train all **(b, c , e, f)** unsupervised approaches on the train splits (without labels). Given a target shape T and 10 segmented train samples, we select T's nearest neighbors S. In Atlasnet **(b, c)**, labels are propagated through the template. In this approach **(e, f, g)**, labels are propagated from T_S to T. We report in (**g**) the best performance of our method over the 10 shots. The mean IoU is reported. Results are averaged over 10 runs.

criterion. The richness of the learned deformations allows our method to find meaningful correspondences in cases where the training example is far from the target shape and the identity baseline does not work. Note that the deformations are often far from isometric. Thus, methods such as 3D-CODED or the approaches of Kanazawa et al. (2018b); Wang et al. (2018a) that rely on regularization toward isometric deformations, would likely fail.

Failure cases. Figure 5.4 shows failures of our method. We show for each category the pair (S,T) which minimizes our segmentation transfer performance. It is clear that the corresponding shapes are rare and specific object instances. We observe two main sources of errors. First, in some cases where we correctly deform S in T, the ground truth labeling was inconsistent, leading to large errors. For example, notice how the source airplane has a single label. Second, S and T are sometimes too distant topologically so that a high-fidelity reconstruction of T is impossible by deforming S. For example, notice how the pole of the lamp has been erroneously inflated to match the target shape.

5.5.2 Quantitative Results

5.5.2.1 Few-shot Segmentation

In this section, we evaluate our approach on the task of transferring semantic labels from a small set of segmented shapes to unlabeled data.

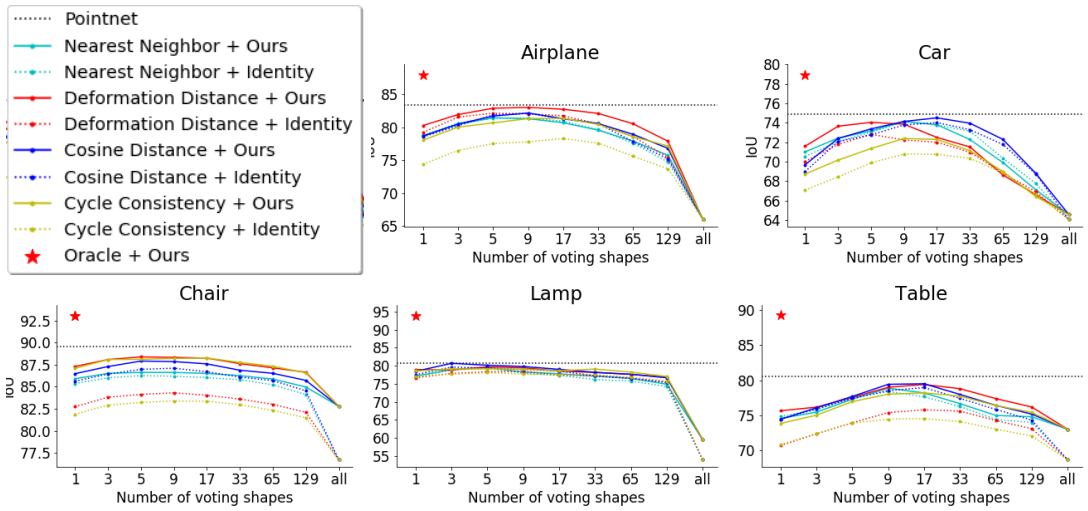


Figure 5.7 Criteria and voting strategies. Study of the number of voting shapes for the transfer of segmentation label, across 4 criteria (see 5.4.3) - Nearest Neighbors, Deformation Distance, Cosine Distance and Cycle Consistency -, and across 5 Shapenet categories. Our transformation method (solid lines) almost always enhance the identity baseline (dashed lines). We report a supervised baseline, Pointnet Qi et al. (2017a) and the oracle source which maximizes IoU for our method. Notice how the oracle significantly outperforms the Pointnet baseline, making the search of a strong selection criterion a good direction. Our models are category specific and trained without segmentation supervision. All of the train set is searched to maximize each criterion.

We report quantitative results for few-shot semantic segmentation on point clouds in Table 5.1. Note that the learning-based methods are all trained separately for each category. Since the results depend on the sampled shapes used in the training set, we report the average and standard deviation over ten randomly sampled training sets. We use the Nearest Neighbors criterion to pair sources and targets and compare our approach against all baselines **(b, c, d, e)**. Notice that our approach out-performs all baselines on all categories. Interestingly, the AtlasNet baseline is not on par with ICP, hinting at the difficulty of predicting two consistent deformations of the template.

We find that the Cycle Consistency criterion **(f)** is a stronger selection criterion than Nearest Neighbors and boosts the results simply by selecting a better (Source, Target) pair. We also report an oracle source-shape selection with our approach where the source shape maximising IoU with the target is selected, which corresponds to the scenario where an optimal source shape is selected. Notice the large improvement of the oracle, showing the quality of our deformations and the potential of our method.

5.5.2.2 Supervised segmentation

Our method is not designed to be competitive when many training samples are available. Indeed, it solves for the deformation against each of the provided segmented shapes, which for large numbers of examples can be computationally expensive compared to feed-forward segmentation predictions like PointNet Qi et al. (2017a). One forward pass through our network deforms a source shape in a target shape in 7 milliseconds (ms), with a 7ms standard deviation (std). ICP takes 28 ms with a 17 std[1]. Here, however, we study the performance of our method in this case, using the segmentation of the many training shapes as supervision during training and making the ten best shapes vote during testing. We report results of our unsupervised method. In addition, we consider adding supervision to our approach by computing Chamfer distances over points with the same segmentation label. The corresponding results are reported in Table 5.2

Table 5.2 shows that, when using all the annotations, nearest neighbors is again a surprisingly good baseline, only slightly below performance of PointNet. Despite the good performance of the identity baseline, our method outperforms it in all categories and performs on par with PointNet. Note that the encoders of our approach incorporate two PointNet architectures, which makes this result intuitive.

Table 5.2 also highlights the importance of the criterion selection. Notice the significant boost in each category gained by carefully choosing the selection criterion over the Nearest Neighbors criterion. The exciting performance of the oracle, way over the PointNet baseline, is another incentive at carefully designing selection criteria.

Finally, notice that our unsupervised trained model is on par with our supervised one. The boost gained by supervised training is marginal except in the car category. It confirms that our cycle-consistent loss is efficient to enforce meaningful part correspondence.

5.5.2.3 Selection criteria and voting strategy

Figure 5.7 shows a quantitative comparison on all criteria, on all category for the identity baseline and our approach using a voting strategy with different number of shapes. The oracle, and PointNet performances are also reported. The Deformation Distance criterion outperforms all other criteria but remains far from the oracle. The oracle performs better than the PointNet baseline across all categories. As a sanity check, we observe that our method outperforms the identity baseline in all settings, showing that it helps to apply our method to transfer labels from S to T.

[1] We use Open3D Zhou et al. (2018) to compute ICP ran on Intel i7-6900K - 3.2 GHz and run our method on an NVIDIA TITAN X.

	Selection	Airplane	Car	Chair	Lamp	Table
(a) Pointnet	-	83.4	74.9	**89.6**	**80.8**	**80.6**
(b) Identity	NN	81.3	74.0	86.1	78.4	78.9
(c) Ours unsup	NN	81.5	73.9	86.6	78.8	79.2
(d) Ours unsup	Best criterion	83.4	74.6	88.4	79.8	79.7
(e) Ours unsup	Oracle	87.9	78.9	93.0	93.9	89.3
(f) Ours sup	NN	81.2	75.9	86.9	78.4	79.0
(g) Ours sup	Best criterion	**83.5**	**76.4**	88.8	79.3	79.9
(h) Ours sup	Oracle	88.0	80.2	93.1	93.4	89.4

Table 5.2 Supervised segmentation:. We compare our approach with **(a)** Pointnet Qi et al. (2017a) and **(b)** Identity baseline. Our approach can be trained with part supervision **(f, g, h)** or without **(c, d, e)**. Given a target shape T and the segmented train set, we compare 3 types of source shapes : **(b, c, f)** T's Nearest Neighbors; **(d, g)** the best shape among all criteria see 5.4.3; and **(e, h)** the *a posteriori* best shape over all train sample. A voting strategy is used on the top 10 shapes in **(b, c, d, f, g)**. The mean IoU is reported.

Figure 5.7 also confirms that using several source shapes is beneficial when many annotated examples are available. In the limit, when all source shapes vote and selection criterion does not matter anymore, an average labelling is predicted with poor performances, which again outlines the importance of source selection. Using nine source shapes performs the best across most criteria and categories when all the training annotations can be used.

5.5.3 Ablation Study

In this section we conduct an ablation study to empirically validate our approach. Table 5.3 shows performances without the cycle loss, without Chamfer loss, and without any specific triplet sampling strategy during training, simply selecting random shapes.

Table 5.3 shows that the cycle consistency loss is critical to the success of our method (relative drop of 23% in IoU). Training without Chamfer distance as a reconstruction loss performs slightly better than the identity baseline and 3% below our approach. This highlight the fact that the cycle consistency loss also acts as a reconstruction loss. Finally, our triplet sampling strategy during training provides a small boost.

Car/100 shots	Nearest Neighbor	Oracle
(a) Identity	67.60	73.59
(b) Ours	**68.19**	**75.87**
(c) Ours w/o cycle loss	52.78	59.63
(d) Ours w/o chamfer	66.21	74.31
(e) Ours w/o knn restriction	67.70	75.23

Table 5.3 Ablation Study:. Given a target shape T and 100 segmented train samples, we select T's nearest neighbors S (1st column), and the oracle source shape which maximizes performances for our approach . (2nd column). We compare **(a)** the identity baseline, with **(b)** our approach, trained without label supervision, and **(c, d, e)** its ablations. The mean IoU is reported. Results are computed on the Car category.

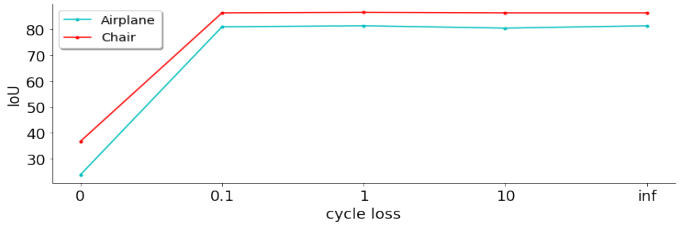


Figure 5.8 Hyperparameter study. Study of the influence of the cycle consistency loss from not having it (abscissa point "0") to having only the cycle loss (abscissa point "inf"). For each target shape, we use the Nearest Neighbors (see 5.4.3) criterion to select sources from the full training set. A voting strategy is used on the top 10 source shapes. The mean IoU is reported

5.5.4 Hyperparameter Study

Figure 5.8 demonstrates once more that the cycle-consistency loss is the pivotal insight of our method. It also outlines the stability of the results for different weightings of our losses. Note how performances are maintained even in the extreme case with only the cycle-consistency loss. Indeed, the identity function is not a trivial minimum of the cycle consistency loss because of the projection step.

5.6 Conclusion

We have extended the correspondence approach of Chapter 4 to arbitrary categories by learning a parametric transformation between two surfaces and leveraging cycle-consistency as a supervisory signal to predict meaningful correspondences. Our method does not require an

object template, can operate without any inter-shape correspondences supervision, and does not assume the deformation is nearly isometric. We demonstrate that our method is able to transfer segmentation labels from a very small number of labeled examples significantly better than state-of-the-art methods, and match the segmentation performance when a larger training dataset is provided.

We believe that the large gap between our performance and the "oracle shape" which provides maximal accuracy shows that using learned deformations to transfer labels, investigating ways to better understand what source models should be selected and new ways to aggregate information across multiple sources is a very promising research direction.

Chapter 6

Conclusion

In this chapter, we summarize our contributions, discuss their impact on the research community and outline research direction they open.

6.1 Contributions

We introduced a new representation for 3D shape, based on surface deformation, and used it to advance the state-of-the-art in single-view reconstruction and shape matching.

- In Chapter 3, we propose an atlas-based modeling of 3D shape. The method learns continuous deformations of a collection of planar patches to reconstruct the surface of a target object. We showcased the strengths of this representation by reconstructing, from a single image, 3D objects spanning 13 categories. Since the learned deformation are continuous, a meshing of the 2D planar patches can be propagated to the 3D surface via the deformation. This makes AtlasNet the first method able to generate a mesh, at arbitrary resolution.

- In Chapter 4, we propose to leverage class information to obtain transformations that respect ground-truth correspondences between shapes. For categories spanned by an underlying template, we propose to deform the template to reconstruct all target shapes and predict dense correspondences between two shapes through the template. The transformations are first learned on a large collection on 3D shapes and then refined at test-time on each new sample with an unsupervised reconstruction objective. Thus is this approach, we use learning to provide a good initialization to a local optimization problem. Our method directly consumes noisy point clouds and we demonstrate it is robust to different types of perturbation. Our approach to shape matching advances the state of the art by 15%.

- In Chapter 5, we extend our method to directly deform any shape into any other shape without using a template. This generalize the previous approach to all categories, even those for which no natural template exists and no annotated correspondences are available for training. Our key insight is to use cycle-consistency to regularize the deformations towards low-distorsion and semantically meaningful deformations. We showcase the strengths and generality of the approach by transferring attributes in shape collection with high intra-category variations.

6.2 Impact

The key take-away from this thesis is that we introduce continuous representations instead of the traditional discretized methods. In all the chapters of this thesis, our purpose was to learn deformations from a reference surface to a target surface, which is modeled either with $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ functions or $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ functions. Previous work applies operators on discretized grids of $\mathbb{R}^2$ or $\mathbb{R}^3$, which is prohibitive in terms of memory consumption when trying to model high-resolution objects. Instead, we simply use MLP to learn continuous approximations of 3D representations. As demonstrated by our experiments, it produces significantly higher quality reconstructions than prior discretized approaches, at just a fraction of the storage cost.

Taking a more precise look a what those MLPs are doing, one can arguy that they simply learn optimised discretization of the input space, like octrees would do but in a hand-crafted way. Indeed, MLPs are piece-wise linear functions and during training, they learns how to discretize the input space by using the ReLU non-linearities adequately in order to best minimize its loss. With that in mind, we conjecture that those learned discretization are superior to hand-crafted regular discretization because they are more efficient: they achieve better results with less memory consumption.

This conjecture was verified in other works as well. In CVPR 2019, three papers use MLPs to encode volumetric representation of objects Chen and Zhang (2019); Mescheder et al. (2019); Park et al. (2019a). In particular, Chen and Zhang (2019); Park et al. (2019a) model the signed distance function and Mescheder et al. (2019) encode the occupancy function. Both the signed distance function and the occupancy function are $\mathbb{R}^3 \rightarrow \mathbb{R}$ functions, and the results of these three papers once again demonstrated that learning the discretization through the internal parameters of a deep neural net was the superior approach. Mildenhall et al. (2020) also demonstrated strong reconstruction results on full 3D scenes using this insight. In their approach, called Nerf, the radiance functions of a 3D scene, a $\mathbb{R}^5 \rightarrow \mathbb{R}^4$ function, is encoded in the weights of an MLP. Their MLP is optimized on a single 3D scene, without learning which shows that this insight holds not only for learned approaches but also optimization methods. This is in line with our obervations in 3D-CODED, where we used learning tp train MLPs and optimized them at test-time on a single 3D shape. Figure 6.1 illustrate the results of Nerf.

All this corpus of work consistently verifies the strength and generality of learning optimized discretization instead of relying on handcrafted ones. This aspect of our contribution is thus a general insight that transfers to other types of data such as image and video.

6.3 The Future

In this section, we discuss three exciting directions to extend the results of the thesis. First, we propose to explore continuous representations of images and videos. Our second direction is to enrich 3D representations with semantic attributes that describe material properties. Lastly, we think that a key to extend the current results to all objects and full 3D scenes is to incorporate structure in the generation process.

6.3.1 Continuous representation for images and videos

This thesis and other works Chen and Zhang (2019); He et al. (2016b); Mescheder et al. (2019); Mildenhall et al. (2020) has established the strengths of MLPs to approximate continuous representation. This general idea could also impact other types of data such as image or video. A continuous representation for images could be an MLP representing an $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ function from pixel space to color space. A video could be continuously represented by a MLP representing a $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ from pixel and time space to color space. First, one obvious advantage of a continuous representation is that is can be sampled at an arbitrary resolution making applications like super-resolution of slow-motion trivial to implement. Second, continuous representations could be useful for compression. Video compression is a major modern tech challenge since about 80% of internet traffic is videos. Deep learning exploiting redundancy in video collections could lead to the next important boost in video compression. Lastly, continuous representation for images and videos also give analytic access to color gradients with regards to time and image location. This could boost image processing methods currently using estimated gradients.

6.3.2 Rich 3D representations

Generating more than spatial geometry is one of the next critical challenges in 3D generation. AtlasNet and later approaches demonstrate that coarse geometry can be generated from a single image. However, all information about texture, material and light propagation is ignored. These properties are essential to create quality 3D assets and are captured implicitly in 2D images. So, moving forward, we want to generate richer 3D representations describing them as well. Ideally, an enriched 3D model has (1) a quad mesh for spatial geometry, (2) a displacement map capturing fine geometric details (3) a texture map for color (4) a bidirectional reflectance distribution function (BRDF) to describe light propagation on the surface. Ideally we would learn high-quality model generation with a dataset of enriched 3D model but such models are

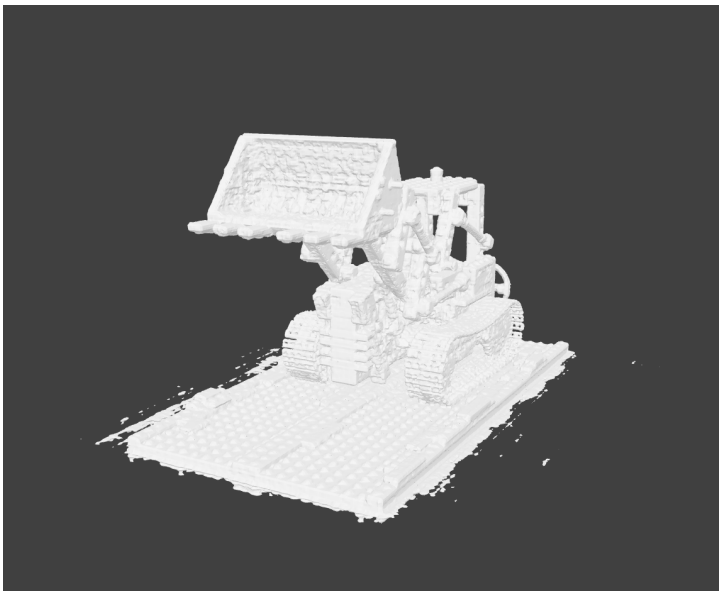


Figure 6.1 Figure from Nerf website Mildenhall et al. (2020). Full 3D scene reconstruction from multi-view images. Nerf remarkably generates more than spatial geometry: high-resolution texture information is implicitly captured in the scene representation. Note this is not explicitly shown by Nerf authors here.

in fact scarce in the public domain. Instead, in the following, we detail two scenarios to learn with multi-view images or videos.

Enriching parametric representations with multi-view images: A first approach to learn enriched model generation is to use several images looking at the same object. Similar to what we did in chapter 5, it is possible to use consistency constraints as supervision. A possible cycle between 2D and 3D representations already proposed in Mildenhall et al. (2020); Sitzmann et al. (2019) in an optimization context is *first 2D image* $\longrightarrow$ *3D model* $\longrightarrow$ *second 2D image*. In this cycle, from a first image a neural net generates a 3D model, which is rendered from the viewpoint of a second image. The rendering matches the second image if the generated 3D model meets two conditions: (1) its spatial geometry must be accurate, (2) its properties - texture, displacement maps and BRDF - must recreate the surface color observed in the second image. Mildenhall et al. (2020) already show exciting results by optimizing a neural net on a single 3D scene, as shown in Figure 6.1. A concrete step to take would be to add the BRDF and the light field to the 3D representations modelled by MLP, simply by augmenting the number of dimensions of the output of the MLP. Indeed, BRDF and light fields are usually smooth and regular, and we previously showed with 3D surfaces that neural networks were very good at parametrizing such families of function. A general advantage of learning 3D generation

with image data is that large-scale image collections are widely available and span much more object categories than 3D datasets currently do. Thus training from images alone would lead to a massive step forward in the diversity of objects we can generate.

Enriching parametric representations with videos: Videos also offer rich 3D signal. Compared to multi-view images, the observed object might have its own dynamic motion. In addition to its inherent material properties, it should be possible to extend 3D generation to model the time dimension of the object. Time being continuous by nature, 3D object dynamics form a smooth family of function, and are good candidates to be modelled by neural 3D mappings. A concrete application would be human tracking in videos, with dense correspondences across time, done by 3D template fitting like in Chapter 4.

6.3.3 Structured generation of 3D geometry

Deep Structured Generation of full 3D scenes: Current deep generation methods struggle to generate structured 3D scenes. Works such as Mildenhall et al. (2020) do reconstruct full 3D scenes from multi-view images but without any structure and not in a learning framework. Structure is also useful for downstream application of scene generation such as animation. One approach to add structure is to view 3D scenes as arrangements of objects. With this assumption, a generation method has to understand *which* objects are present and *where* they are. Our initial experiments show that even when restricting ourselves to the minimal synthetic examples of random arrangement of 3D spheres, AtlasNet fails to perform satisfactory reconstructions. The main problem in trying to map a set of parts to a target is predicting the *occurrence* of each part, because it is a discrete notion, hardly differentiable. Not only improvement in this area could have a large impact in scaling generation methods from objects to scenes, but we find this to be a common problem appearing in other works: Li et al. (2018a); Paschalidou et al. (2019) cast occurrence prediction as supervised deep learning problem, but reinforcement learning might be a better way to look at it since it is a discrete decision problem Tulsiani et al. (2016). To gain insights on this problem, one possible approach is to lift in 3D the 2D region proposal methods (e.g., Faster RCNN Ren et al. (2015)) since they give compelling results on a similar problem for images. A first concrete step in that direction was taken by Gkioxari et al. (2019) by adding a mesh decoder branch to a mask-rcnn architecture, in order to reconstruct a 3D scene of each detected object in an image.

Deep Structured Generation to all 3D objects: Interestingly, objects are to scenes what elementary parts are to objects. As for scenes, we think generating 3D object with a part structure is a key step for generalization. Learning parts has the additional long-term advantage

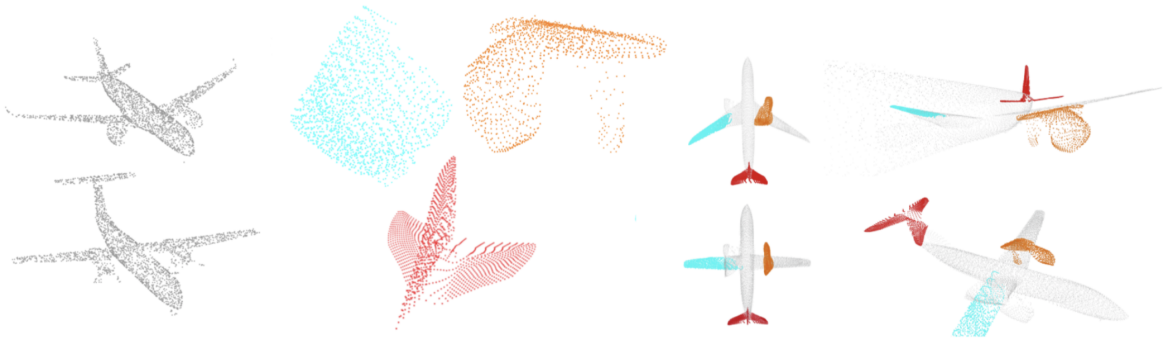


Figure 6.2 Primitive-based reconstructions. Figure from Deprelle et al. (2019). Qualitative visualizations of automatically learned primitives (called "learned elementary structures") for shape reconstruction and matching from real-world examples. This work complements the approach presented in chapter 3.

of enabling part-based editing of a 3D asset from a human editor. We believe in the following insight: while there are thousands of object categories and modeling each of them is intractable, all man-made objects are essentially made of the same parts. For example, an algorithm able to generate 3D chairs by assembling elementary parts should generalize to tables as those categories are made of the same elementary parts. We have already taken concrete steps in this direction. In our NeurIPS publication, Theo Drepelle showed that the elementary parts can be learned Deprelle et al. (2019). To learn part shapes, the main idea is that template-based deformation methods such as AtlasNet or 3D-CODED from chapter 3 and 4 are end-end-end differentiable with regard the template they deform. One possibility to learn parts is thus to consider the template points as learnable parameters of AtlasNet and 3D-CODED. Deprelle et al. (2019) also propose another possibility to learn parts by learning their surface deformation from an initial simple shape, and using them template in AtlasNet or 3D-CODED. Excitingly, both approaches lead to clear improvement in shape generation and shape matching which goes to show that learning parts is an important problem. Perhaps even more exciting, the learned parts tend to have semantic meaning even though no semantic supervision was used during training. Figure 6.2 shows discovered parts on the plane category including a shape reactor, a wing and a plane tail.

Appendix A

Additional Results on AtlasNet

This appendix provides more detailed quantitative and qualitative results highlighting the strengths and limitations of AtlasNet.

A.1 Detailed results, per category

These tables report the metro reconstruction error and the chamfer distance error. It surprisingly shows that our method with 25 learned parameterizations outperforms our method with 125 learned parameterizations in 7 categories out of 13 for the metro distance, but is significantly worse on the cellphone category, resulting in the 125 learned parameterizations approach being better on average. This is not mirrored in the Chamfer distance.

	pla.	ben.	cab.	car	cha.	mon.	lam.	spe.	fir.	cou.	tab.	cel.	wat.	mean
Baseline PSR	2.71	2.12	1.98	2.24	2.68	1.78	2.58	2.29	1.03	1.90	2.66	1.15	2.46	2.12
Baseline PSR PA	1.38	1.97	1.75	2.04	2.08	1.53	2.51	2.25	1.46	1.57	2.06	1.15	1.80	1.82
Ours 1 patch	1.11	1.41	1.70	1.93	1.76	1.35	2.01	2.30	1.01	1.46	1.46	0.87	1.46	1.53
Ours 1 sphere	1.03	1.33	1.64	1.99	1.76	1.30	2.06	2.33	0.93	1.41	1.59	0.79	1.54	1.52
Ours 5 patch	0.99	1.36	1.65	**1.90**	1.79	1.28	2.00	2.27	0.92	**1.37**	1.57	**0.76**	1.40	1.48
Ours 25 patch	**0.96**	1.35	1.63	1.96	1.49	**1.22**	**1.86**	**2.22**	0.93	1.36	**1.31**	1.41	**1.35**	1.47
Ours 125 patch	1.01	**1.30**	**1.58**	**1.90**	**1.36**	1.29	1.95	2.29	**0.85**	1.38	1.34	**0.76**	1.37	**1.41**

Table A.1 Auto-Encoder (per category). The mean is taken category-wise. The Metro Distance is reported, multiplied by 10. The meshes were contructed by propagating the patch grid edges.

	pla.	ben.	cab.	car	cha.	mon.	lam.	spe.	fir.	cou.	tab.	cel.	wat.	me
Baseline	1.11	1.46	1.91	1.59	1.90	2.20	3.59	3.07	0.94	1.83	1.83	1.71	1.69	1.
Baseline + normal	1.25	1.73	2.19	1.74	2.19	2.52	3.89	3.51	0.98	2.13	2.17	1.87	1.88	2.
Ours 1 patch	1.04	1.43	1.79	2.28	1.97	1.83	3.06	2.95	0.76	1.90	1.95	1.29	1.69	1.
Ours 1 sphere	0.98	1.31	2.02	1.75	1.81	1.83	2.59	2.94	0.69	1.73	1.88	1.30	1.51	1.
Ours 5 patch	0.96	1.21	**1.64**	1.76	1.60	**1.66**	2.51	**2.55**	0.68	1.64	1.52	**1.25**	1.46	1.
Ours 25 patch	0.87	1.25	1.78	1.58	1.56	1.72	2.30	2.61	0.68	1.83	1.52	1.27	1.33	1.
Ours 125 patch	**0.86**	**1.15**	1.76	**1.56**	**1.55**	1.69	**2.26**	**2.55**	**0.59**	**1.69**	**1.47**	1.31	**1.23**	**1.**

Table A.2 Auto-Encoder (per category). The mean is taken category-wise. The Chamfer Distance is reported, multiplied by 10^3.

		pla.	ben.	cab.	car	cha.	mon.	lam.	spe.	fir.	cou.	tab.	cel.	wat.	mean
metro	HSP	1.10	1.84	1.28	1.06	1.61	1.66	1.93	1.77	1.05	1.37	1.93	1.39	1.34	1.49
	Ours 25 patch	**0.77**	**1.01**	**1.04**	**0.92**	**1.19**	**1.22**	**1.26**	**1.46**	**0.95**	**1.19**	**1.27**	**0.83**	**1.09**	**1.09**
chamfer	HSP	2.60	17.4	14.3	1.77	10.0	19.4	9.46	21.7	2.34	12.9	20.2	13.2	4.89	11.6
	Ours 25 patch	**1.33**	**14.1**	**12.5**	**1.29**	**7.23**	**17.5**	**6.99**	**17.8**	**1.69**	**11.2**	**17.0**	**10.6**	**4.20**	**9.52**

Table A.3 Single-view reconstruction. Quantitative comparison against HSP Häne et al. (2017), a state of the art octree-based method. The average error is reported, on 100 shapes from each category. The Chamfer Distance reported is computed on 10^4 points, and multiplied by 10^3. The Metro distance is multiplied by 10.

A.2 Regularisation

In the autoencoder experiment, we tried using weight decay with different weight. The best results were obtained without any regularization.

Weight Decay	Ours : 25 patches
10^{-3}	8.57
10^{-4}	4.84
10^{-5}	3.42
0	1.56

Table A.4 Regularization on Auto-Encoder (per category). The mean is taken category-wise. The Chamfer Distance is reported, multiplied by 10^3.

A.3 Additional Single View Reconstruction qualitative results

In this figure, we show one example of single-view reconstruction per category and compare with the state of the art, PointSetGen and 3D-R2N2. We consistently show that our method produces a better reconstruction.

A.4 Additional Autoencoder qualitative results

In this figure, we show one example per category of autoencoder reconstruction for the baseline and our various approaches to reconstruct meshes, detailed in the main paper. We show how we are able to recreate fine surfaces.

A.5 Additional Shape Correspondences qualitative results

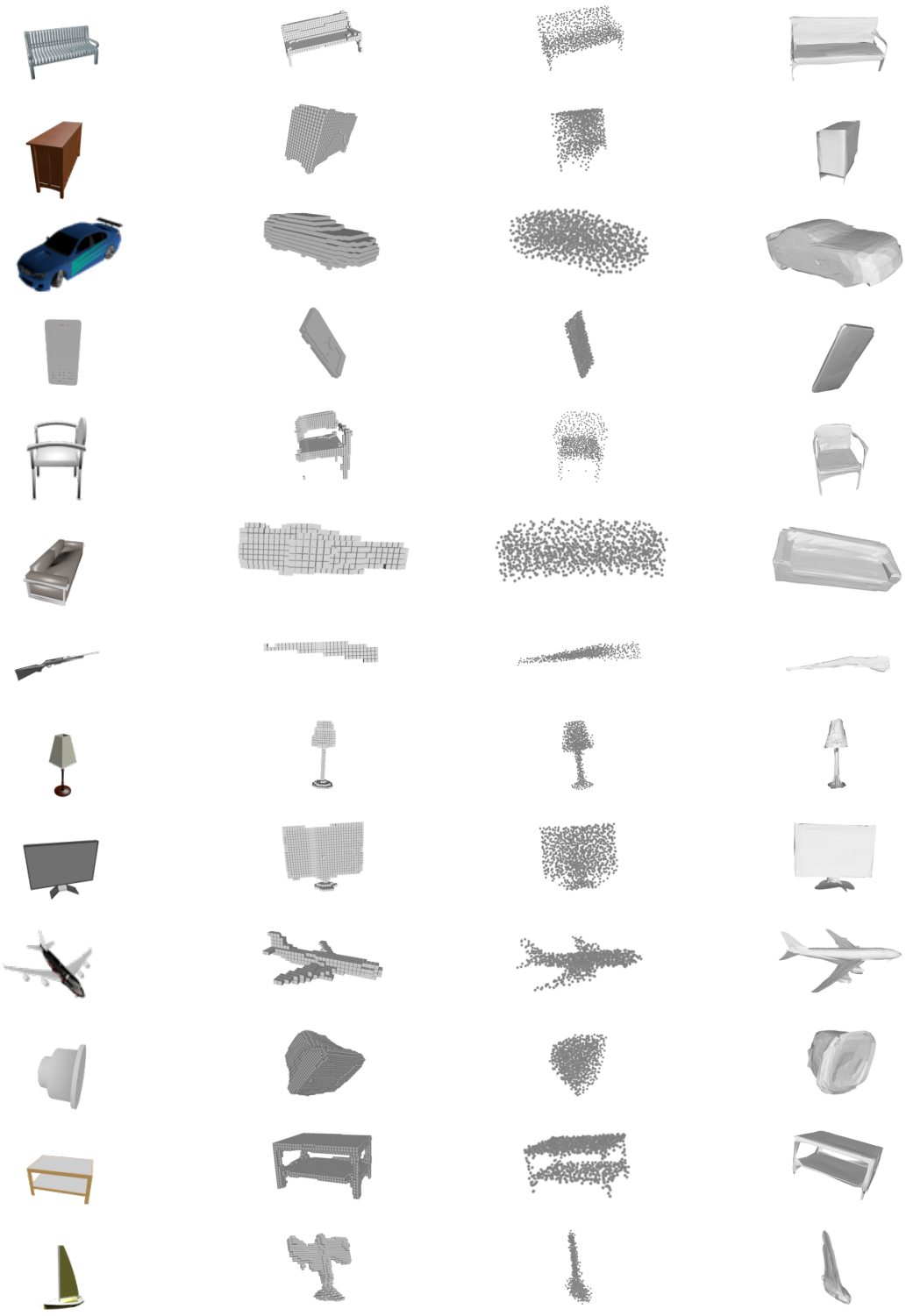
We color each vertex of the reference object by its distance to the gravity center of the object, and transfer these colors to the inferred atlas. We then propagate them to other objects of the same category, showing semantically meaningful correspondences between them. Results for the plane and watercraft categories are shown and generalize to all categories.

A.6 Deformable shapes.

We ran an experiment on human shape to show that our method is also suitable for reconstructing deformable shapes. The FAUST dataset Bogo et al. (2014) is a collection of meshes representing several humans in different poses. We used 250 shapes for training, and 50 for validation (without using the ground truth correspondences in any way). In table A.5, we report the reconstruction error in term of Chamfer distance and Metro distance for our method with 25 squarred parameterizations, our methods with a sphere parametrization, and for the baseline. We found results to be consistent with the analysis on ShapeNet. Qualitative results are shown in figure A.5, revealing that our method leads to qualitatively good reconstructions.

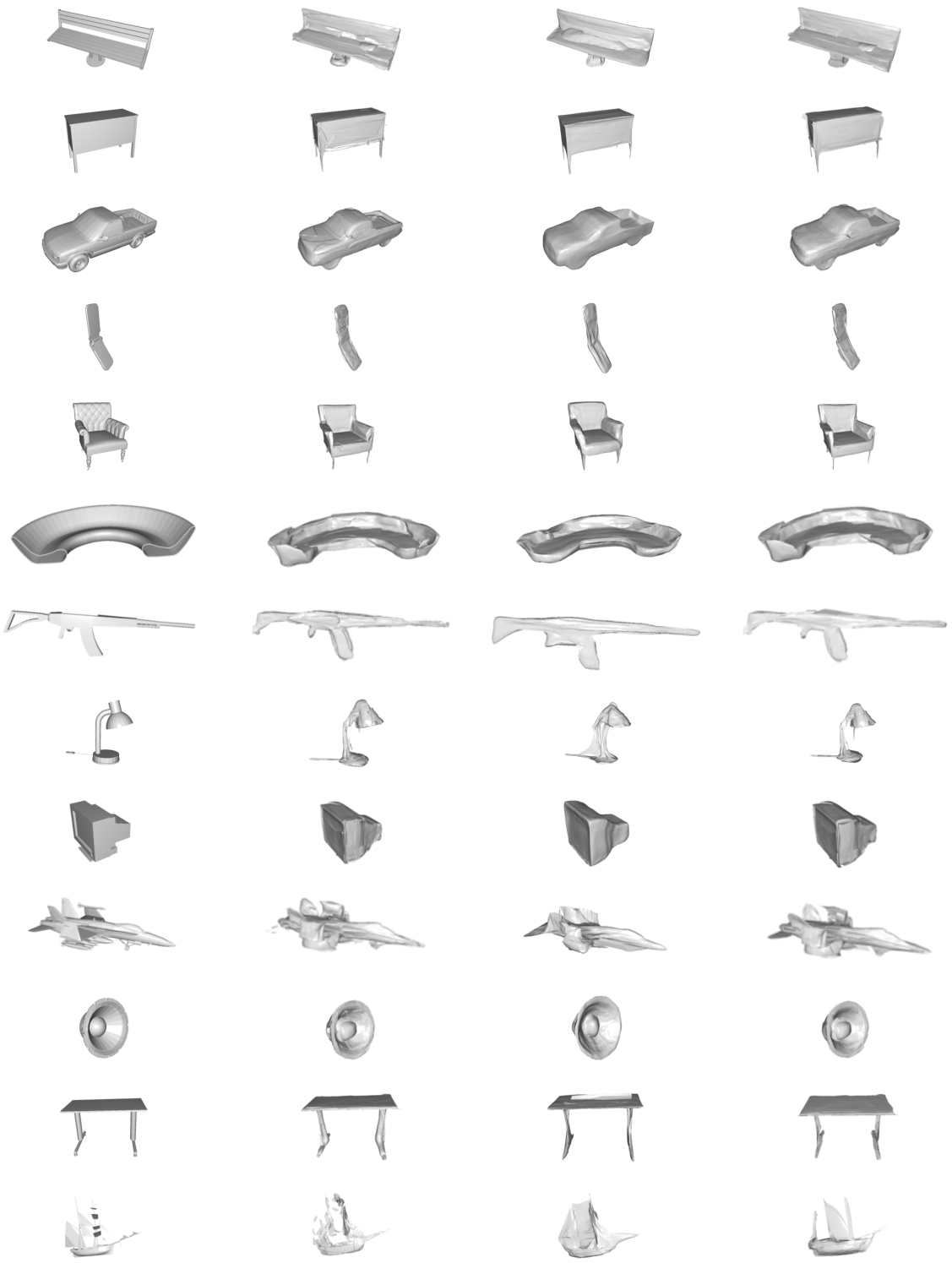
A.7 Point cloud super-resolution

AtlasNet can generate pointclouds or meshes of arbitrary resolution simply by sampling more points. Figure A.6 shows qualitative results of our approach with 25 patches generating high



(a) input **(b)** 3D-R2N2 **(c)** PSG **(d)** Ours

Figure A.1 Single-view reconstruction comparison: From a 2D RGB image (a), 3D-R2N2 reconstructs a voxel-based 3D model (b), PointSetGen a point cloud based 3D model (c), and our AtlasNet a triangular mesh (d).



(a) Ground truth **(b)** PSR on ours **(c)** Ours sphere **(d)** Ours 25

Figure A.2 Autoencoder comparison: We compare the original meshes (a) to meshes obtained by running PSR (b) on the dense point cloud sampled from our generated mesh, and to our method generating a surface from a sphere (c), and 25 (d) learnable parameterizations.

pt

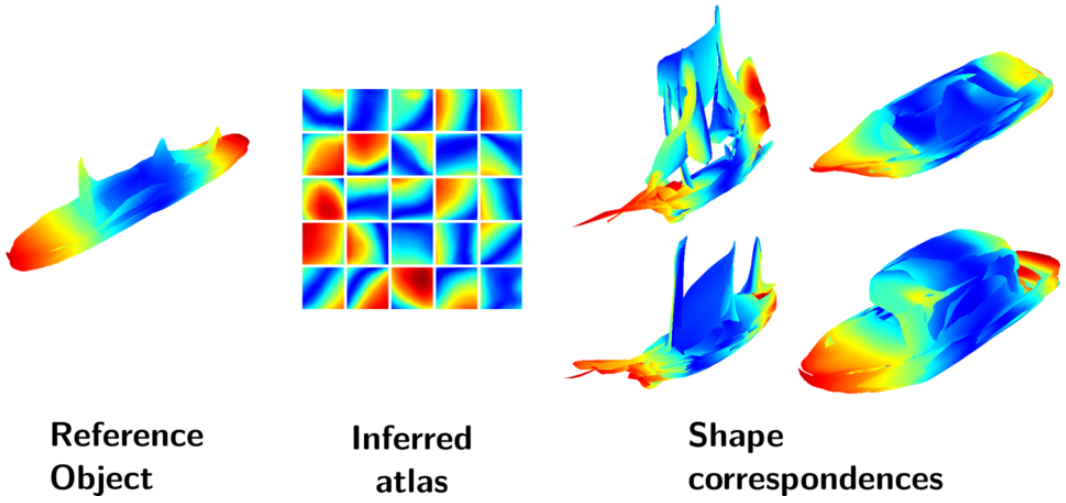


Figure A.3 Shape correspondences: a reference watercraft (left) is colored by distance to the center, with the jet colormap. We transfer the surface colors to the inferred atlas for the reference shape (middle). Finally, we transfer the atlas colors to other shapes (right). Notice that we get semantically meaningful correspondences, without any supervision from the dataset on semantic information. All objects are generated by the autoencoder, with 25 learned parametrizations.

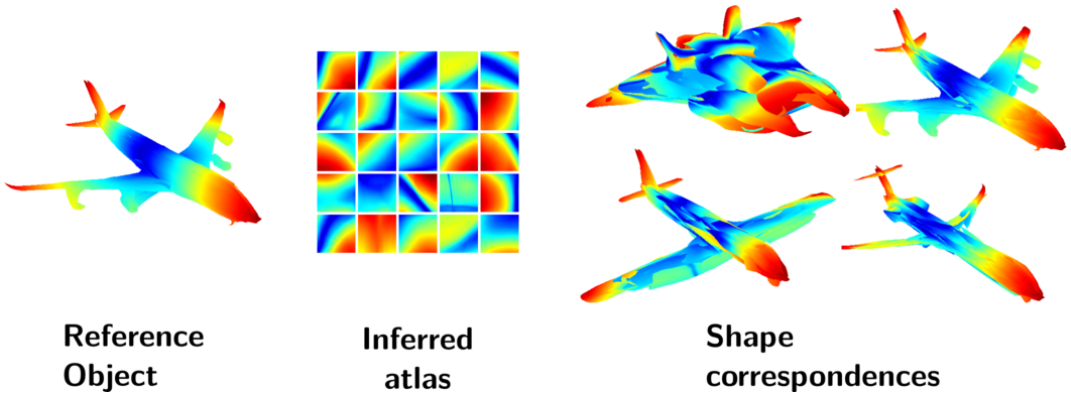


Figure A.4 Shape correspondences: a reference plane (left) is colored by distance to the center, with the jet colormap. We transfer the surface colors to the inferred atlas for the reference shape (middle). Finally, we transfer the atlas colors to other shapes (right). Notice that we get semantically meaningful correspondences, such as the nose and tail of the plane, and the tip of the wings, without any supervision from the dataset on semantic information. All objects are generated by the autoencoder, with 25 learned parametrizations.

Figure A.5 Deformable shapes. Our method learned on 250 shapes from the FAUST dataset to reconstructs a human in different poses. Each color represent one of the 25 parametrizations.

	Chamfer	Metro
25 patches	15.47	11.62
1 Sphere	15.78	15.22
1 Ref. Human	16.39	13.46

Table A.5 3D Reconstruction on FAUST Bogo et al. (2014). We trained the baseline and our method sampling the points according from 25 square patches, and from a sphere on the human shapes from the FAUST dataset. We report Chamfer distance (x 10^4) on the points and Metro distance (x10) on the meshes.

(a) Low-Res Input **(b)** High-Res reconstruction

Figure A.6 Super resolution. Our approach can generate meshes at arbitrary resolutions, and the pointnet encoder Qi et al. (2017a) can take pointclouds of varying resolution as input. Given the same shape sampled at the training resolution of 2500, or 10 times less points, we generate high resolution meshes with 122500 vertices. This can be viewed as the 3D equivalent of super-resolution on 2D pixels.

resolution meshes with 122500 points. Moreover, PointNet is able to take an arbitrary number of points as input and encodes a minimal shape based on a subset of the input points. This is a double-edged sword : while it allows the autoencoder to work with varying number of input points, it also prevent it from reconstructing very fine details, as they are not used by PointNet and thus not present in the latent code. We show good results using only 250 input points, despite the fact that we train using 2500 input points which shows the capacity of our decoder to interpolate a surface from a small number of input points, and the flexibility of our pipeline.

A.8 Details on the comparison against HSP Häne et al. (2017)

We perform a quantitative comparison against an octree-based state of the art method. AtlasNet is trained with 25 learned parameterizations on the same data as their publicly available trained model[1]. 100 random samples are drawn from each category from the test split. We evaluated the the quality of the reconstruction using the Chamfer distance on the unnormalized meshes, and the metro distance. Voxelised versions of meshes often appear inflated. This bias can appear for HSP, where we observed that the generated meshes were slightly larger than the

[1] https://github.com/chaene/hsp.

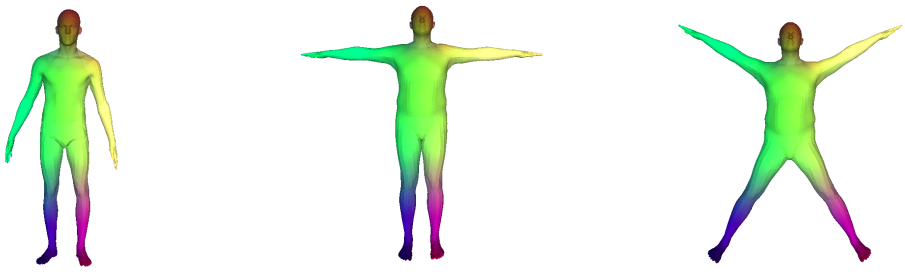
original meshes. We ran an ICP alignment procedure on the generated meshes for both methods to remove this bias. In table A.3, we report per category results. As AtlasNet was specifically trained to optimise the chamfer distance, we outperform HSP in every category. AtlasNet also outperforms HSP in metro distance in each category for the metro distance, for which none of the two algorithm where trained to optimise. List of sampled used, ans trained model for AtlasNet are available in the github repository.

Appendix B

Additional Results on 3D-CODED

B.1 Choice of template

The template is a critical element for our method. We experimented with three different templates: (i) a "FAUST" template associated with SMPL parameters fitted to a body in a neutral pose in the FAUST training set, (ii) a "zero" template corresponding to the "zero" shape of SMPL, and (iii) a "separated" template in which this "zero" shape is modified to have the legs better separated and the arms higher. In this experiment, the points are not sampled regularly on the surface, and a low resolution template is used. Figure B.1 shows the different templates, while table B.1 shows quantitative results using the different templates. Interestingly, the best results were obtained with the more "natural" template, selected in the "FAUST" training dataset, rather than with the templates from simple SMPL parameters, where points from different body parts seem easier to separate.



(a) "FAUST" template **(b)** "Zero" template **(c)** "Separated" template

Figure B.1 Shapes for template study. We evaluate three different template shapes used in our model.

template 0	Faust error (cm)
"FAUST" template	**3.255**
"Zero" template	3.385
"Separated" template	3.314

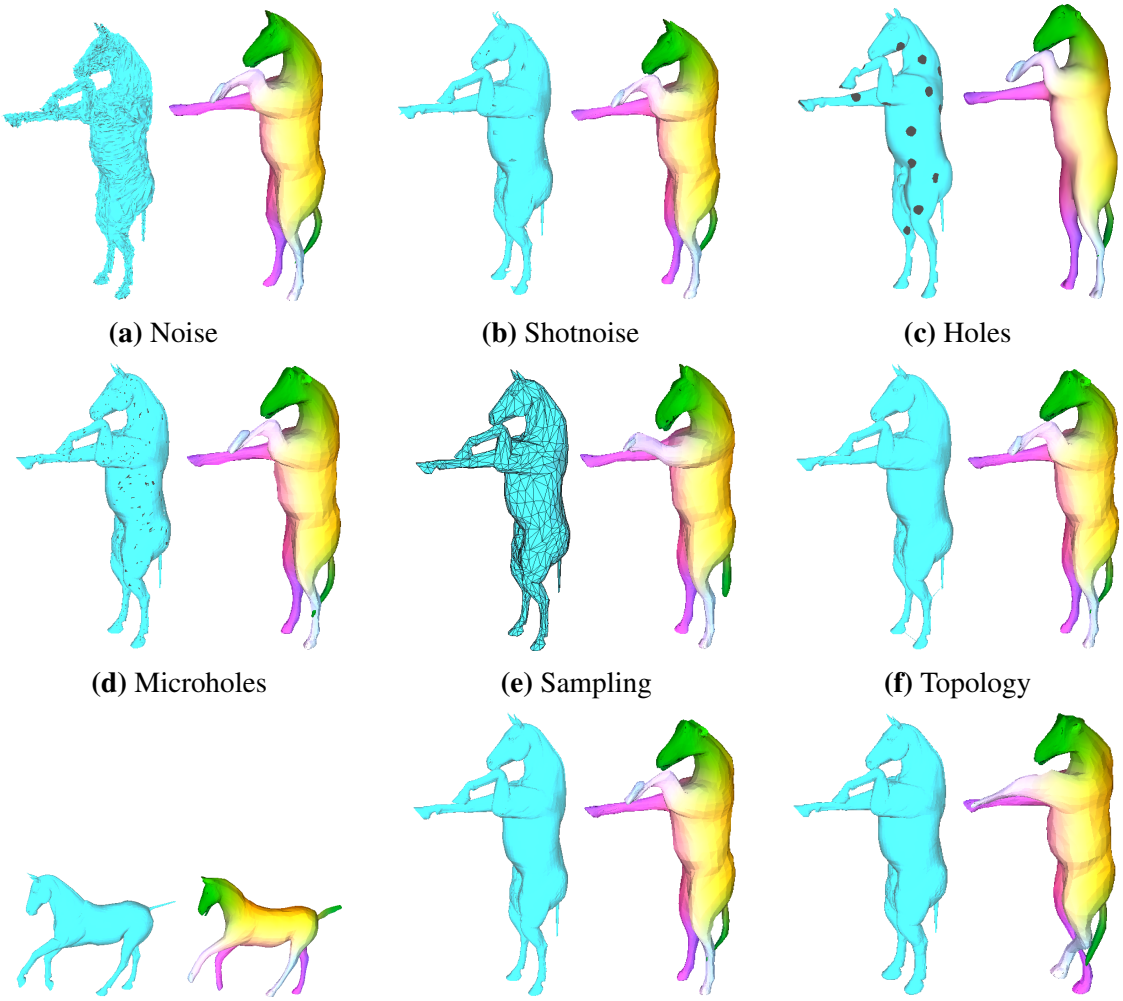
Table B.1 Comparison of different template shapes. We compare different choices for the template shape shown in Figure B.1. Notice that the neutral "FAUST" template performs best out of the three tested shapes.

Perturbation		Error (cm)	Perturbation		Error (cm)	Perturbation		Error (cm)
Noise	1	4.58	Scale	1	4.73	Holes	1	4.71
	2	3.87		2	4.78		2	4.71
	3	3.93		3	4.66		3	4.72
	4	3.67		4	4.62		4	4.69
	5	3.91		5	4.67		5	4.84
ShotNoise	1	4.66	Local scale	1	4.18	Microholes	1	4.71
	2	2.64		2	3.65		2	4.72
	3	3.03		3	3.62		3	4.82
	4	2.72		4	3.75		4	4.69
	5	3.00		5	3.56		5	3.53
Sampling	1	4.82	Topology	1	3.99	Isometry	1	4.72
	2	4.78		2	4.38		2	4.69
	3	4.61		3	4.37		3	4.79
	4	3.72		4	4.31		4	4.85
	5	9.93		5	7.53		5	4.74

Table B.2 Quantitative results for perturbations on TOSCA for the horse category

B.2 Quantitative results for perturbations on TOSCA

We quantitatively evaluate the robustness of our method to perturbation on the TOSCA dataset. This dataset consists of several versions of the same synthetic mesh with different perturbations, specifically: noise, shotnoise, sampling, scale, local scale, topology, holes, microholes, and isometry. We experimented on the horse model. In Table B.2 we report quantitative results for each perturbation (with a gradual strength from 1 to 5) and show qualitative reconstruction with correspondences suggested by colors for each category with maximum strength in Figure B.2. We found that we are robust to all categories of noise under study, except for strong variation in sampling (964 points instead of 19948) Surprisingly, adding noise can enhance the quantitative error.



(a) Noise **(b)** Shotnoise **(c)** Holes

(d) Microholes **(e)** Sampling **(f)** Topology

(g) Isometry **(h)** Scale **(i)** Local scale

Figure B.2 Robustness to perturbations on TOSCA for the horse category. Correspondences are suggested by color. Notice the overall robustness to all perturbations, with small errors on the ears, tail or legs.

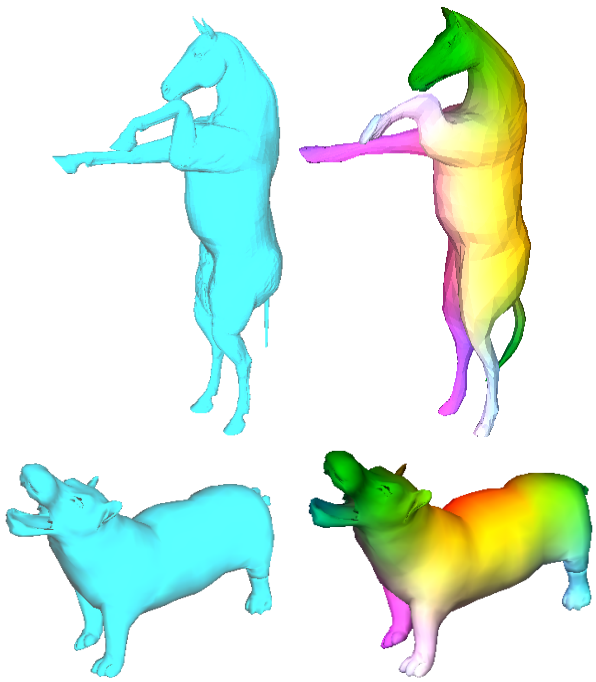


Figure B.3 Inter-class correspondences on animals. Correspondences are suggested by color.

B.3 Cross-category correspondances on animals

SMAL synthetic are in correspondences across categories. Hence the template for two different categories are in correspondence and our approach can be trivially extended to get correspondences for animals from different species. Qualitative evidence of this is show in Figure B.3.

B.4 Regularization for the unsupervised case

In the unsupervised case of equation 4.2, if the autoencoder is trained using the Chamfer distance alone, it falls into a bad local minimum with high distorsion of the template to reconstruct the input shape. For example the left foot is propagated on left hand in Figure 4.10. This distortion is consistent across shapes, so correspondences are still possible, and perform reasonably well with an average error of 8.727cm on the FAUST-inter challenge. However, we expect that by minimizing distorsion in the generated shape, the *Shape Deformation Network* will learn to map an arm to an arm, and a foot to a foot, which will naturally encourage correspondences. We added two regularization losses to achieve this: an edge loss $\mathcal{L}^{\text{edges}}$ and a laplacian loss $\mathcal{L}^{\text{Lap}}$.

B.4.1 Edge loss $\mathcal{L}^{\text{edges}}$

Let (V,E) be the graph of the template and V^r the reconstructed vertices.

$$\mathcal{L}^{\text{edges}}(V^r) = \frac{1}{\#E} \cdot \sum_{(i,j)\in E} \mid \frac{\|V_i^r - V_j^r\|}{\|V_i - V_j\|} - 1 \mid \tag{B.1}$$

This enforces edges to keep the same length in the template and the generated mesh. We use $\mathcal{L}^{\text{edges}} = 0.005$. For instance, if the length of an edge doubles the contribution to the loss is $\mathcal{L}^{\text{edges}} \cdot 1.0 = 0.005$ which is equivalent (in terms of contribution to the loss function) to a error of placement of 7.1cm. In other words, in terms on loss for the network, it is equivalent to double an edge's length or to misplace a point by 3.2cm.

B.4.2 Laplacian loss $\mathcal{L}^{\text{Lap}}$

Similar to Kanazawa et. al. **?**, we use the Laplacian regularization. The Laplacian matrix L is defined as :

$$L_{i,j} = \begin{cases} d_i \text{ if } i = j \\ -1 \text{ if } (i,j) \in E \\ 0 \text{ otherwize} \end{cases} \tag{B.2}$$

$$\begin{aligned} [LV]_i &= \sum_{(i,j)\in E} V_i - V_j \\ &= d_i \cdot (V_i - \frac{\sum_{(i,j)\in E} V_j}{d_i}) \end{aligned} \tag{B.3}$$

This is an approximation of the following integral as explained in Sorkine (2006).

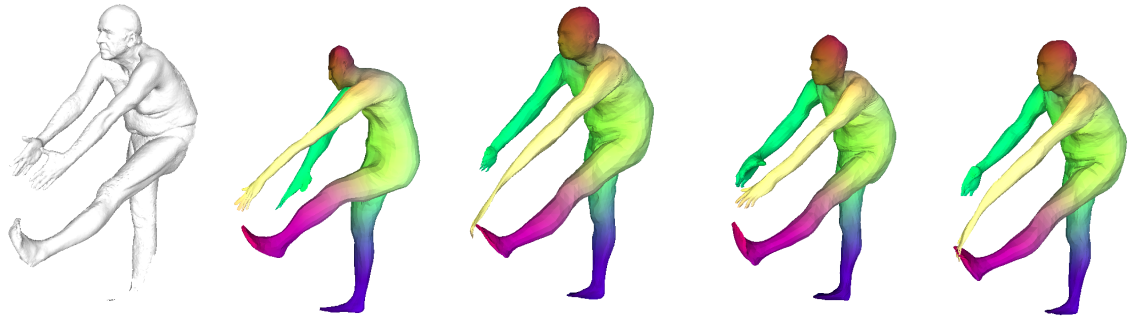
$$\lim_{\gamma \to 0} \frac{1}{\mid \gamma \mid} \int_{v\in\gamma} (v_i - v) dl(v) = -H(v_i) \cdot n_i \tag{B.4}$$

where:

- $H(v_i)$ is the mean curvature
- n_i is the surface normal

We follow Meyer et al. (2001) and use cotangent weights in the Laplacian which have been shown to have better geometric discretization.

$$[L^c V]_i = \frac{1}{\Omega_i} \sum_{i\sim j} \frac{1}{2} (cot\alpha_{ij} + cot\beta_{ij})(V_i - V_j) \tag{B.5}$$



(a) target T **(b)** Result R **(c)** R attracts T **(d)** T attracts R **(e)** Both ways

Figure B.4 Asymmetric Chamfer loss in reconstruction optimization. Given an input scan, with holes (a), our network outputs a reconstruction result (b), that can be improved by an optimization step. When the scan has holes, it is better to only consider a loss where the scan attracts the reconstruction (d), rather than using a loss where reconstruction attracts the scan (c), or the Chamfer distance where they attract each other (e).

where :

- Ω_i is the size of the Voronoi cell of i
- α_{ij} and β_{ij} denote the two angles opposite of edge (i, j)

Our Laplacian loss is thus written :

$$\mathcal{L}^{\text{Lap}}(V^g) = \mathbb{1}^t \cdot L^c \cdot (V^{template} - V^r) \tag{B.6}$$

We use $\lambda_{laplace} = 0.005$. In practice we notice that using Laplacian regularization constrains the network to keep sound surfaces. It may still suffer from error in symmetry and can still invert right and left, and front and back.

B.5 Asymmetric Chamfer distance

Figure B.4 illustrates that optimizing an asymmetric Chamfer distance can in some cases, especially when the 3D scans have holes, produce qualitatively better results. However, Table B.3 shows that the symmetric version of the Chamfer distance performs better. Investigating how other losses behave, such the Earth-Mover distance loss (also known as Wasserstein loss) behave is left to future work.

Method	Faust error (cm)
Without regression	6.29
With regression, Chamfer asym (R attracts T)	4.023
With regression, Chamfer asym (T attracts R)	3.336
With regression (both ways)	3.255

Table B.3 Analysis on the Chamfer distance. We compare the latent feature search with Chamfer Distance against latent feature searches with asymmetric Chamfer distances. On average, the Chamfer distance (symmetric) performs better (no regular sampling on the surface, low-resolution template).

Bibliography

Adelson, E. H. and Pentland, A. P. (1996). The perception of shading and reflectance. In *Perception as Bayesian Inference*. Cambridge University Press.

Aflalo, Y., Brezis, H., and Kimmel, R. (2014). On the optimality of shape and data representation in the spectral domain.

Allen, B., Curless, B., and Popovic, Z. (2002). Articulated body deformation from range scan data. In *SIGGRAPH*.

Allen, B., Curless, B., and Popovic, Z. (2003). The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH*.

Allen, B., Curless, B., and Popovic, Z. (2006). Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis. In *Symposium on Computer Animation*.

Amberg, B., Romdhani, S., and Vetter, T. (2007). Optimal step nonrigid icp algorithms for surface registration. *2007 IEEE Conference on Computer Vision and Pattern Recognition*.

Andreux, M., Rodola, E., Aubry, M., and Cremers, D. (2014). Anisotropic laplace-beltrami operators for shape analysis. In *Sixth Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*.

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). Scape: shape completion and animation of people. In *ACM transactions on graphics (TOG)*.

Aubry, M., Schlickewei, U., and Cremers, D. (2011). The wave kernel signature: A quantum mechanical approach to shape analysis. In *IEEE International Conference on Computer Vision (ICCV) - Workshop on Dynamic Shape Capture and Analysis (4DMOD)*.

Azencot, O., Corman, E., Ben-Chen, M., and Ovsjanikov, M. (2017). Consistent functional cross field design for mesh quadrangulation. In *ACM Trans. Graph.*

Barron, J. T. and Malik, J. (2015). Shape, illumination, and reflectance from shading.

Barrow, H., Tenenbaum, J., Hanson, A., and Riseman, E. (1978). Recovering intrinsic scene characteristics.

Basri, R. and Jacobs, D. (2001). Photometric stereo with general, unknown lighting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*.

Bednarík, J., Salzmann, M., and Fua, P. (2020). Learning to reconstruct texture-less deformable surfaces from a single view. In *CVPR*.

Belhumeur, P. N., Kriegman, D. J., and Yuille, A. L. (1999). The bas-relief ambiguity. Springer.

Belongie and Malik (2000). Matching with shape contexts. In *2000 Proceedings Workshop on Content-based Access of Image and Video Libraries*.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts.

Besl, P. and McKay, H. (1992). A method for registration of 3-d shapes. ieee trans pattern anal mach intell. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Besl, P. J., McKay, N. D., et al. (1992). A method for registration of 3-d shapes.

Biederman, I. (1985). Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. In *Psychological review*.

Binford, I. (1971). Visual perception by computer. In *IEEE Conference of Systems and Control*.

Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*.

Bo Li, Chunhua Shen, Yuchao Dai, van den Hengel, A., and Mingyi He (2015). Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., and Black, M. J. (2016). Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Computer Vision – ECCV 2016*.

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

Bonneel, N., Kovacs, B., Paris, S., and Bala, K. (2017). Intrinsic decompositions for image editing.

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

Boscaini, D., Masci, J., Rodola, E., and Bronstein, M. M. (2016a). Learning shape correspondence with anisotropic convolutional neural networks. In *NIPS*.

Boscaini, D., Masci, J., Rodolà, E., Bronstein, M. M., and Cremers, D. (2016b). Anisotropic diffusion descriptors. In *Computer Graphics Forum*.

Bouaziz, S. and Pauly, M. (2013). Dynamic 2d/3d registration for the kinect. In *ACM SIGGRAPH 2013 Courses*.

Brady, M. and Yuille, A. (1984). An extremum principle for shape from contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Brock, A., Donahue, J., and Simonyan, K. (2019). Large scale gan training for high fidelity natural image synthesis. In *ICLR*.

Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2006a). Efficient computation of isometry-invariant distances between surfaces. In *SIAM J. Scientific Computing*.

Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2006b). Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. In *Proc. National Academy of Sciences (PNAS)*.

Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2008). Numerical geometry of non-rigid shapes.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. In *IEEE Signal Processing Magazine*.

Bronstein, M. M. and Kokkinos, I. (2010). Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Brown, B. J. and Rusinkiewicz, S. (2007). Global non-rigid alignment of 3-d scans. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

Brown, M., Hua, G., and Winder, S. (2011). Discriminative learning of local image descriptors. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bruna, J., Zaremba, W., Szlam, A., and Lecun, Y. (2014). Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR)*.

Brunton, A., Salazar, A., Bolkart, T., and Wuhrer, S. (2014). Review of statistical shape spaces for 3d data with comparative analysis for human faces. *Computer Vision and Image Understanding*.

Carroll, R., Ramamoorthi, R., and Agrawala, M. (2011). Illumination decomposition for material recoloring with consistent interreflections. *ACM Trans. Graph.*

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). ShapeNet: An Information-Rich 3D Model Repository. In *Arxiv*.

Chen, Q. and Koltun, V. (2015). Robust nonrigid registration by convex optimization. In *International Conference on Computer Vision (ICCV)*.

Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*.

Chen, Z. and Zhang, H. (2019). Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *eccv*.

Cignoni, P., Rocchini, C., and Scopigno, R. (1998). Metro: Measuring error on simplified surfaces. In *Computer Graphics Forum*.

Corona, E., Pumarola, A., Alenya, G., Moreno-Noguer, F., and Rogez, G. (2020). Ganhand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*.

Delanoy, J., Aubry, M., Isola, P., Efros, A. A., and Bousseau, A. (2018). 3d sketching using multi-view deep volumetric prediction. In *Proceedings of the ACM on Computer Graphics and Interactive Techniques*.

Deprelle, T., Groueix, T., Fisher, M., Kim, V., Russell, B., and Aubry, M. (2019). Learning elementary structures for 3d shape generation and matching. In *Advances in Neural Information Processing Systems*, pages 7433–7443.

DeTone, D., Malisiewicz, T., and Rabinovich, A. (2017). Superpoint: Self-supervised interest point detection and description.

D.Raviv, A.Dubrovina, and R.Kimmel (2013). Hierarchical framework for shape correspondence. In *Numerical Mathematics: Theory, Methods and Applications*.

Duchêne, S., Riant, C., Chaurasia, G., Moreno, J. L., Laffont, P.-Y., Popov, S., Bousseau, A., and Drettakis, G. (2015). Multiview intrinsic images of outdoors scenes with an application to relighting. *ACM Trans. Graph.*

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., and Sattler, T. (2019). D2-net: A trainable cnn for joint detection and description of local features. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Dyke, R. M., Lai, Y.-K., Rosin, P. L., and Tam, G. K. (2019a). Non-rigid registration under anisotropic deformations. In *Computer Aided Geometric Design*.

Dyke, R. M., Stride, C., Lai, Y.-K., Rosin, P. L., Aubry, M., Boyarski, A., Bronstein, A. M., Bronstein, M. M., Cremers, D., Fisher, M., Groueix, T., Guo, D., Kim, V. G., Kimmel, R., Lähner, Z., Li, K., Litany, O., Remez, T., Rodolà, E., Russell, B. C., Sahillioglu, Y., Slossberg, R., Tam, G. K. L., Vestner, M., Wu, Z., and Yang, J. (2019b). Shape Correspondence with Isometric and Non-Isometric Deformations. In Biasotti, S., Lavoué, G., and Veltkamp, R., editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association.

Eigen, D. and Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *2015 IEEE International Conference on Computer Vision (ICCV)*.

Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*.

Ekman, P. and Friesen, W. (1978). Facial action coding system: A technique for the measurement of facial movement. In *Consulting Psychologists Press*.

Elad, A. and Kimmel, R. (2003). On bending invariant signatures for surfaces.

Fan, H., Su, H., and Guibas, L. (2017). A point set generation network for 3D object reconstruction from a single image. In *cvpr*.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.

Fu, H., Gong, M., Wang, C., Batmanghelich, K., and Tao, D. (2018). Deep ordinal regression network for monocular depth estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Gehler, P. V., Rother, C., Kiefel, M., Zhang, L., and Schölkopf, B. (2011). Recovering intrinsic images with a global sparsity prior on reflectance. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset.

Girdhar, R., Fouhey, D., Rodriguez, M., and Gupta, A. (2016). Learning a predictable and generative vector representation for objects. In *eccv*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*.

Gkioxari, G., Malik, J., and Johnson, J. (2019). Mesh r-cnn. In *ICCV*.

Godard, C., Mac Aodha, O., and Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *CVPR*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. http://www.deeplearningbook.org.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*.

Graves, A., Mohamed, A., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*.

Groueix, T., Fisher, M., Kim, V. G., Russell, B., and Aubry, M. (2018a). AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Groueix, T., Fisher, M., Kim, V. G., Russell, B., and Aubry, M. (2018b). Supplementary material (appendix) for the paper https://http://imagine.enpc.fr/~groueixt/atlasnet/arxiv.

Gu, X., Gortler, S., and Hoppe, H. (2002). Geometry images. In *SIGGRAPH*.

Halimi, O., Litany, O., Rodola, E., Bronstein, A. M., and Kimmel, R. (2019). Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Han, X., Li, Z., Huang, H., Kalogerakis, E., and Yu, Y. (2017). High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *iccv*.

Häne, C., Tulsiani, S., and Malik, J. (2017). Hierarchical surface prediction for 3D object reconstruction. In *3DV*.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition.

Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., and Schmid, C. (2019a). Learning joint reconstruction of hands and manipulated objects. In *CVPR*.

Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., and Schmid, C. (2019b). Learning joint reconstruction of hands and manipulated objects. In *CVPR*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hodan, T., Kouskouridas, R., Kim, T.-K., Tombari, F., Bekris, K., Drost, B., Groueix, T., Walas, K., Lepetit, V., Leonardis, A., Steger, C., Michel, F., Sahin, C., Rother, C., and Matas, J. (2018). A summary of the 4th international workshop on recovering 6d object pose. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.

Hormann, K., Polthier, K., and Sheffer, A. (2008). Mesh parameterization: Theory and practice. In *ACM SIGGRAPH ASIA 2008 Courses*.

Horn, B. K. (1974). Determining lightness from an image.

Horn, B. K. P. (1975). Obtaining shape from shading information. In *The Psychology of Computer Vision*.

Horn, B. K. P. and Brooks, M. J. (1989). Shape from shading. In *MIT Press*.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. In *Neural networks*.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Huang, Q.-X. and Guibas, L. (2013). Consistent shape maps via semidefinite programming. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*.

Huang, Q.-X., Zhang, G.-X., Gao, L., Hu, S.-M., Butscher, A., and Guibas, L. (2012). An optimization approach for extracting and encoding consistent maps in a shape collection. In *ACM Trans. Graph.*

Ikeuchi, K. and Horn, B. K. (1981). Numerical shape from shading and occluding boundaries.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. (2017). Image-to-image translation with conditional adversarial networks. In *cvpr*.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. In *Advances in neural information processing systems*.

Jatavallabhula, K. M., Smith, E., Lafleche, J.-F., Fuji Tsang, C., Rozantsev, A., Chen, W., Xiang, T., Lebaredian, R., and Fidler, S. (2019). Kaolin: A pytorch library for accelerating 3d deep learning research. *arXiv:1911.05063*.

Jinggang Huang, Lee, A. B., and Mumford, D. (2000). Statistics of range images. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*.

Johnson, A. E. (1997). Spin-images: a representation for 3-d surface matching.

Johnson, A. E. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. In *IEEE Transactions on pattern analysis and machine intelligence*.

Kalogerakis, E., Averkiou, M., Maji, S., and Chaudhuri, S. (2017). 3D shape segmentation with projective convolutional networks. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*.

Kalogerakis, E., Hertzmann, A., and Singh, K. (2010). Learning 3d mesh segmentation and labeling. In *ACM Transactions on Graphics (TOG)*.

Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018a). End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Regognition (CVPR)*.

Kanazawa, A., Tulsiani, S., Efros, A. A., and Malik, J. (2018b). Learning category-specific mesh reconstruction from image collections. In *ECCV*.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. In *ACM Transactions on Graphics (TOG)*.

Kim, V. G., Li, W., Mitra, N. J., DiVerdi, S., and Funkhouser, T. (2012). Exploring Collections of 3D Models using Fuzzy Correspondences. In *Transactions on Graphics (Proc. of SIGGRAPH)*.

Kim, V. G., Lipman, Y., and Funkhouser, T. (2011). Blended Intrinsic Maps. In *Transactions on Graphics (Proc. of SIGGRAPH)*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., and Panozzo, D. (2019). Abc: A big cad model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Koenderink, J. (1984). What does the occluding contour tell us about solid shape? *Perception*.

Koenderink, J. J., van Doorn, A. J., Christou, C., and Lappin, J. S. (1996). Shape constancy in pictorial relief. In Ponce, J., Zisserman, A., and Hebert, M., editors, *Object Representation in Computer Vision II*. Springer Berlin Heidelberg.

Kokkinos, I., Bronstein, M. M., Litman, R., and Bronstein, A. M. (2012). Intrinsic shape context descriptors for deformable shapes.

Kolev, K., Tanskanen, P., Speciale, P., and Pollefeys, M. (2014). Turning mobile phones into 3d scanners. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Kovalsky, S. Z., Galun, M., and Lipman, Y. (2016). Accelerated quadratic proxy for geometric optimization. In *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.

Kwatra, V., Han, M., and Dai, S. (2012). Shadow removal for aerial imagery by information theoretic intrinsic image analysis. In *2012 IEEE International Conference on Computational Photography (ICCP)*. IEEE.

Körtgen, M., Park, G.-J., Novotni, M., and Klein, R. (2003). 3d shape matching with 3d shape contexts.

Land, E. H. and McCann, J. J. (1971). Lightness and retinex theory. OSA.

Lewis, J. P., Cordner, M., and Fong, N. (2000). Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*.

Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., and Guibas, L. (2017). GRASS: Generative recursive autoencoders for shape structures. In *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*.

Li, K., Yang, J., Lai, Y., and Guo, D. (2019). Robust non-rigid registration with reweighted position and transformation sparsity. In *IEEE Transactions on Visualization and Computer Graphics*.

Li, L., Sung, M., Dubrovina, A., Yi, L., and Guibas, L. (2018a). Supervised fitting of geometric primitives to 3d point clouds. In *CVPR*.

Li, Y., Wang, G., Ji, X., Xiang, Y., and Fox, D. (2018b). Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Li, Z. and Snavely, N. (2018). Learning intrinsic image decomposition from watching the world. In *Computer Vision and Pattern Recognition (CVPR)*.

Lin, C.-H., Wang, O., Russell, B. C., Shechtman, E., Kim, V. G., Fisher, M., and Lucey, S. (2019). Photometric mesh optimization for video-aligned 3d object reconstruction. In *CVPR*.

Lipman, Y. and Funkhouser, T. (2009). Mobius voting for surface correspondence. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

Litany, O., Remez, T., Rodola, E., Bronstein, A. M., and Bronstein, M. M. (2017). Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*.

Litman, R. and Bronstein, A. (2013). Learning spectral descriptors for deformable shape correspondence.

Liu, X., Wan, L., Qu, Y., Wong, T.-T., Lin, S., Leung, C.-S., and Heng, P.-A. (2008). Intrinsic colorization. *ACM Trans. Graph.*

Loper, M., Mahmood, N., Romero, J., Pons-Moll, and G., Black, M. J. (2015). Smpl: A skinned multi-person linear model. In *SIGGRAPH Asia*.

Luo, Z., Shen, T., Zhou, L., Zhang, J., Yao, Y., Li, S., Fang, T., and Quan, L. (2019). Contextdesc: Local descriptor augmentation with cross-modality context.

Ma, W.-C., Chu, H., Zhou, B., Urtasun, R., and Torralba, A. (2018). Single image intrinsic decomposition without a single intrinsic image. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*.

Mamassian, P., Kersten, D., and Knill, D. C. (1996). Categorical local-shape perception. *Perception*.

Maron, H., Galun, M., Aigerman, N., Trope, M., Dym, N., Yumer, E., Kim, V. G., and Lipman, Y. (2017). Convolutional neural networks on surfaces via seamless toric covers. In *SIGGRAPH*.

Masci, J., Boscaini, D., Bronstein, M. M., and Vandergheynst, P. (2015). Geodesic convolutional neural networks on riemannian manifolds. In *Proc. of the IEEE International Conference on Computer Vision (ICCV) Workshops*.

Mémoli, F. and Sapiro, S. (2005). A theoretical and computational framework for isometry invariant recognition of point cloud data. In *Foundations of Computational Mathematics*.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*.

Meyer, M., Desbrun, M., Schr, P., and Barr, A. (2001). Discrete differential-geometry operators for triangulated 2-manifolds. In *Proceedings of Visualization and Mathematics*.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *CVPR*.

Mitra, N. J., Wand, M., Zhang, H., Cohen-Or, D., Kim, V. G., and Huang, Q.-X. (2014). Structure-Aware Shape Processing. In *SIGGRAPH Course notes*.

Mo, K., Zhu, S., Chang, A., Yi, L., Tripathi, S., Guibas, L., and Su, H. (2019a). PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mo, K., Zhu, S., Chang, A. X., Yi, L., Tripathi, S., Guibas, L. J., and Su, H. (2019b). PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Monnier, T., Groueix, T., and Aubry, M. (2020). Deep transformation-invariant clustering. *arXiv:2006.11132*.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., and Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*.

Mundy, J. (2006). Object recognition in the geometric era: A retrospective.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*.

Nguyen, A., Ben-Chen, M., Welnicka, K., Ye, Y., and Guibas, L. (2011). An optimization approach to improving collections of shape maps. In *Computer Graphics Forum*.

Omer, I. and Werman, M. (2004). Color lines: image specific color representation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*

Opelt, A., Fussenegger, M., Pinz, A., and Auer, P. (2004). Weak hypotheses and boosting for generic object detection and recognition. In Pajdla, T. and Matas, J., editors, *Computer Vision - ECCV 2004*.

Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., and Guibas, L. (2012). Functional maps: A flexible representation of maps between shapes. In *ACM Trans. Graph.*

Ovsjanikov, M., Sun, J., and Guibas, L. (2008). Global intrinsic symmetries of shapes.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019a). Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*.

Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019b). Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Paschalidou, D., Gool, L., and Geiger, A. (2020). Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Paschalidou, D., Ulusoy, A. O., and Geiger, A. (2019). Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., and Black, M. J. (2019). Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Pentland, A. (1984). Local shading analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

Pentland, A. (1986). Parts: Structured descriptions of shape. In *AAAI*.

Prados, E.and Faugeras, O. (2006). Shape from shading. In *Handbook of Mathematical Models in Computer Vision*. Springer US.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). PointNet: Deep learning on point sets for 3D classification and segmentation. In *cvpr*.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *nips*.

R3DS, W. . (2018). Russian3dscanner: Wrap 3.3. In *https://www.russian3dscanner.com/*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Neurips*.

Revaud, J., Weinzaepfel, P., de Souza, C. R., and Humenberger, M. (2019). R2D2: repeatable and reliable detector and descriptor. In *NeurIPS*.

Riegler, G., Ulusoy, A. O., Bischof, H., and Geiger, A. (2017). OctNetFusion: Learning depth fusion from data. In *3DV*.

Roberts, L. G. (1963). Machine perception of three-dimensional solids.

Rodola, E., Rota Bulo, S., Windheuser, T., Vestner, M., and Cremers, D. (2014). Dense non-rigid shape correspondence using random forests. In *CVPR*.

Rodolà, E., Bronstein, A. M., Albarelli, A., Bergamasco, F., and Torsello, A. (2012). A game-theoretic approach to deformable shape matching.

Romero, J., Tzionas, D., and Black, M. J. (2017). Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain.

Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. (2006). 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*.

Rubner, Y., Tomasi, C., and Guibas, L. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*.

Ruo Zhang, Ping-Sing Tsai, Cryer, J. E., and Shah, M. (1999). Shape-from-shading: a survey.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*.

Russell, B. C., Sivic, J., Ponce, J., and Dessales, H. (2011). Automatic alignment of paintings and photographs depicting a 3d scene. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*.

Rustamov, R. M. (2007). Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*.

Rustamov, R. M., Ovsjanikov, M., Azencot, O., Ben-Chen, M., Chazal, F., and Guibas, L. (2013). Map-based exploration of intrinsic shape differences and variability.

Sahillioglu, Y. and Yemez, Y. (2011). Coarse-to-fine combinatorial matching for dense isometric shape correspondence. In *Computer Graphics Forum*.

Sahillioğlu, Y. (2018). A genetic isometric shape correspondence algorithm with adaptive sampling. In *ACM Trans. Graph.*

Sander, P., Wood, Z., Gortler, S., Snyder, J., and Hoppe, H. (2003). Multi-chart geometry images. In *SGP*.

Saxena, A., Sun, M., and Ng, A. Y. (2009). Make3d: Learning 3d scene structure from a single still image. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play.

Simonovsky, M. and Komodakis, N. (2017). Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sinha, A., Bai, J., and Ramani, K. (2016). Deep learning 3d shape surfaces using geometry images. In *CVPR*.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. (2019). Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*.

Sorkine, O. (2006). Differential representations for mesh processing. In *Comput. Graph. Forum*.

Su, W., Zhang, H., Li, J., Yang, W., and Wang, Z. (2019). Monocular depth estimation as regression of classification using piled residual networks.

Sun, J., Ovsjanikov, M., and Guibas, L. (2009a). A concise and provably informative multi-scale signature-based on heat diffusion". In *Computer Graphics Forum (Proc. of SGP)*.

Sun, J., Ovsjanikov, M., and Guibas, L. (2009b). A concise and provably informative multi-scale signature based on heat diffusion.

Takuya Narihira, M. M. and Yu, S. X. (2015). Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *International Conference on Computer Vision (ICCV)*.

Tam, G. K. L., Cheng, Z., Lai, Y., Langbein, F. C., Liu, Y., Marshall, D., Martin, R. R., Sun, X., and Rosin, P. L. (2013). Registration of 3d point clouds and meshes: A survey from rigid to nonrigid.

Tatarchenko, M., Dosovitskiy, A., and Brox, T. (2017). Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *iccv*.

Tatarchenko, M., Richter, S. R., Ranftl, R., Li, Z., Koltun, V., and Brox, T. (2019). What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tombari, F., Salti, S., and Stefano, L. D. (2010). Unique signatures of histograms for local surface description. In *ECCV*.

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle adjustment — a modern synthesis. In Triggs, B., Zisserman, A., and Szeliski, R., editors, *Vision Algorithms: Theory and Practice*. Springer Berlin Heidelberg.

Tulsiani, S., Su, H., Guibas, L. J., Efros, A. A., and Malik, J. (2016). Learning shape abstractions by assembling volumetric primitives. In *Computer Vision and Pattern Recognition (CVPR)*.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. In *Arxiv*.

van Kaick, O., Zhang, H., Hamarneh, G., and Cohen-Or, D. (2011). A survey on shape correspondence.

Varol, G., Ceylan, D., Russell, B., Yang, J., Yumer, E., Laptev, I., and Schmid, C. (2018). BodyNet: Volumetric inference of 3D human body shapes. In *ECCV*.

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017). Learning from synthetic humans. In *CVPR*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, u., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.

Vestner, M., Lähner, Z., Boyarski, A., Litany, O., Slossberg, R., Remez, T., Rodola, E., Bronstein, A., Bronstein, M., Kimmel, R., and Cremers, D. (2017). Efficient deformable shape correspondence via kernel matching. In *2017 International Conference on 3D Vision (3DV)*.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A. J., Chung, J., Choi, D. H., Powell, R. W., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T. P., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G. (2018a). Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., and Tong, X. (2017). O-cnn: Octree-based convolutional neural networks for 3d shape analysis. In *ACM Transactions on Graphics (SIGGRAPH)*.

Wang, W., Ceylan, D., Mech, R., and Neumann, U. (2019a). 3dn: 3d deformation network. In *CVPR*.

Wang, X. and Phillips, C. (2002). Multi-weight enveloping: Least-squares approximation techniques for skin animation.

Wang, X., Zhou, B., Shi, Y., Chen, X., Zhao, Q., and Xu, K. (2019b). Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *CVPR*.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2018b). Dynamic graph cnn for learning on point clouds. In *ACM Trans. Graph.*

Weinzaepfel, P. and Rogez, G. (2019). Mimetics: Towards understanding human actions out of context. *arXiv preprint arXiv:1912.07249*.

Wen, C., Zhang, Y., Li, Z., and Fu, Y. (2019). Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE International Conference on Computer Vision*.

Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., and Panozzo, D. (2019). Deep geometric prior for surface reconstruction. In *CVPR*.

Woodford, O. J., Torr, P. H. S., Reid, I. D., and Fitzgibbon, A. W. (2008). Global stereo reconstruction under second order smoothness priors. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*.

Woodham, R. (1992). Photometric method for determining surface orientation from multiple images. volume 19.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. (2019). A comprehensive survey on graph neural networks.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Xiao, Y., Qiu, X., Langlois, P.-A., Aubry, M., and Marlet, R. (2019). Pose from shape: Deep pose estimation for arbitrary 3d objects. *arXiv preprint arXiv:1906.05105*.

Xu, Q., Wang, W., Ceylan, D., Mech, R., and Neumann, U. (2019). Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*.

Yang, Y., Feng, C., Shen, Y., and Tian, D. (2018). Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*.

Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., and Guibas, L. (2016a). A scalable active framework for region annotation in 3d shape collections. In *SIGGRAPH Asia*.

Yi, L., Su, H., Guo, X., and Guibas, L. (2016b). Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *CVPR*.

Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Yong Chang, J., Mu Lee, K., Molchanov, P., Kautz, J., Honari, S., Ge, L., Yuan, J., Chen, X., Wang, G., Yang, F., Akiyama, K., Wu, Y., Wan, Q., Madadi, M., Escalera, S., Li, S., Lee, D., Oikonomidis, I., Argyros, A., and Kim, T.-K. (2018). Depth-based 3d hand pose estimation: From current achievements to future goals. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zagoruyko, S. and Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zaharescu, A., Boyer, E., Varanasi, K., and Horaud, R. (2009). Surface feature detection and description with applications to mesh matching. In *CVPR*.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces.

Zhou, Q.-Y., Park, J., and Koltun, V. (2018). Open3D: A modern library for 3D data processing.

Zhou, T., Krähenbühl, P., Aubry, M., Huang, Q., and Efros, A. A. (2016). Learning dense correspondence via 3d-guided cycle consistency. In *Computer Vision and Pattern Recognition (CVPR)*.

Zhou, T., Krähenbühl, P., and Efros, A. A. (2015). Learning data-driven reflectance priors for intrinsic image decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *iccv*.

Zollhöfer, M., Thies, J., Garrido, P., Bradley, D., Beeler, T., Pérez, P., Stamminger, M., Nießner, M., and Theobalt, C. (2018). State of the art on monocular 3d face reconstruction, tracking, and applications. *Computer Graphics Forum*.

Zuffi, S. and Black., M. J. (2015). The stitched puppet: A graphical model of 3d human shape and pose. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Zuffi, S., Kanazawa, A., and Black, M. J. (2018). Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zuffi, S., Kanazawa, A., Jacobs, D., and Black, M. J. (2017). 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*.