Rheinische
Friedrich-Wilhelms-
Universität Bonn

Institute for Computer Science
Department VI
Autonomous Intelligent Systems

Rheinische
Friedrich-Wilhelms-Universität Bonn

Master thesis

Latent Space Non-Rigid Registration Method for Learning Class-Level Object Shape Transformations

Author:
Corbin Cogswell

First Examiner:
Prof. Dr. Sven Behnke

Second Examiner:
Prof. Dr. Reinhard Klein

Advisor:
Ph.D. Seongyong Koo

Submitted: June 19, 2017

Declaration of Authorship

I declare that the work presented here is original and the result of my own investigations. Formulations and ideas taken from other sources are cited as such. It has not been submitted, either in part or whole, for a degree at this or any other university.

Location, Date

Signature

Abstract

Robots working in human environments must be able to use a variety of tools and objects. Using these tools requires manipulation information such as grasp poses or trajectories. However, even within the same class, objects express some variation in their shape. This means that each individual instance of an object must be trained or manually annotated with its own manipulation information. It is often time-consuming or impractical to program or train a robot for every instance of an object. By using registration methods, a transformation between a previously-trained shape and a newly-observed shape can be found, and manipulation information from the previously-trained instance can be transferred to the observed instance, facilitating the observed object's use. Previous methods work by directly registering the two shapes, but these methods are sensitive to the shapes' initial alignments and lack robustness to noise, occlusions, and self-occlusions. Because these methods lack class-level knowledge of the object, there is no way to infer what parts of an observed object are noise or occlusions nor can hidden parts be inferred. To overcome these limitations, this thesis proposes a method for non-rigid registration using class-level information combined with subspace methods. Rather than directly searching the entire space of transformations between two shapes, the proposed method first defines a canonical shape and finds a set of transformations which deform it to match other known instances of the same class. Then PCA is applied to this set to find a lower-dimensional latent space of transformations. Inference on novel fully- or partially-observed objects is done by minimizing an energy function by concurrently searching the latent space of transformations and fitting a single rigid transformation. By restricting the search space around transformations relevant to the class of objects, the proposed registration method increases the robustness to many of the problems of previous methods and gains some ability to infer hidden or self-occluded parts. This thesis will demonstrate the registration accuracy of our method on fully- and partially-observed shapes and demonstrate its robustness to poor initial alignment, noise, and occlusions.

Acknowledgements

I sincerely thank Dr. Koo for his absolutely invaluable help and support. I would also like to thank Professor Behnke for his suggestion of this topic and his help in making it a reality and Professor Klein for his help in reviewing this thesis. I thank Michael Schreiber, Oliver Burghard, and Arul Selvam for providing meshes and data to use during the development and testing of the thesis. And finally, I would like to thank my friends and family who made this thesis possible.

Contents

List of Figures

List of Algorithms

1. Introduction

1.1. Motivation

A human can be given a screwdriver they have never observed before, and they will immediately know how to grasp and operate it. The handle might be wider, the shank might be shorter, and there can be other variations in the shape, but the human will still instantly know where to grasp it and how to operate it. A human does not need to re-learn how to use every instance of a tool; they can simply adapt any previous manipulation knowledge to the changes in shape of the new instance and transfer over this manipulation information. This allows them to immediately make use of a novel instance of a tool just as well as they could with any previous and more familiar instances.

And while such a feat is trivial for humans, this is not the case for robots. The variance of shapes of objects even within the same class is often enough that previous manipulation knowledge a robot may have may not be immediately applicable to a novel instance of an object. The robot must therefore learn or be taught how to use every instance of a tool, regardless of any previous experience it may have with other instances from the same class. This can be time-consuming, it can require human interaction, and, in many scenarios, it can be simply impossible.

Rather than just discarding the manipulation information a robot may have from a similar object, the manipulation knowledge can be modified to be applicable to the novel instance. Stueckler et al. (2011) proposed a method for manipulation skill transfer using non-rigid registration. The registration finds a non-rigid transformation between a prior instance where manipulation information - such as grasp poses or motion trajectories - was available and a novel instance from the same class. The transformation is given as a dense deformation field: i.e. for any point in the space of the first shape, the deformation field gives a vector which transforms that point into the space of the second. For any manipulation knowledge defined in the space of the first instance, the deformation field is applied to this knowledge as well, transforming it into the space of the newly observed instance and facilitating the object's use.

However, there are practical issues which affect the results of non-rigid registration. There are problems such as noise from sensors, occlusions from camera

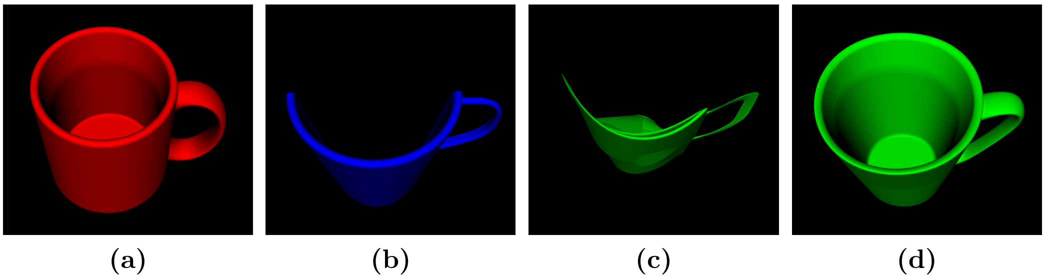


Figure 1.1.: A prior template shape (red) matched against a partially-occluded target observed shape (blue) and the resulting registration matches (green). Left: Coherent Point Drift (c), right: our method (d). Notice that while the general model (CPD) has collapsed the occluded far wall in to match with the observed wall, our learned method maintains a mug-like shape more consistent with the true shape of the partially-hidden object.

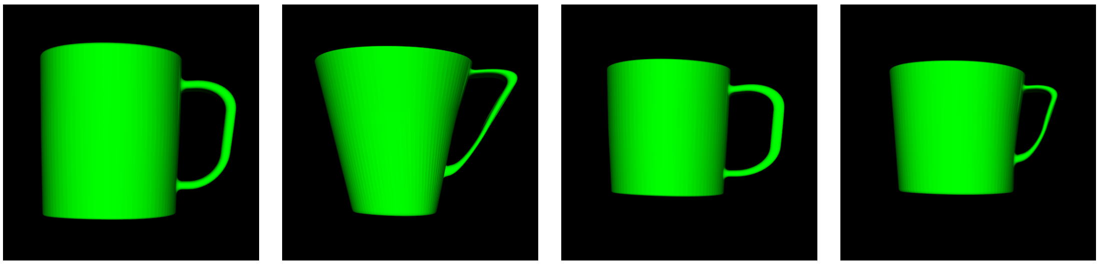


Figure 1.2.: Novel samples of mugs from a latent space of mug transformations using linear combinations of previously observed transformations.

perceptive or from other objects in the scene (such as the robot's own manipulators), and initial misalignments between the template and target shapes. These problems can lead to incorrect correspondences between points and incorrect transformations. In particular, occlusions lead to missing data, which can be hard or even impossible to infer from the point data alone. For methods which rely on the accuracy of the registration results, this can be very problematic.

While a general model has no concept of the objects being registered and thus can only rely on heuristics such as smoothness or motion coherence, a learned model can leverage information based on previously observed transformations between other objects of the same class. We propose a learned non-rigid registration method which is robust to these issues by incorporating class-level information. Our method automatically finds and learns the transformations which are most common to a class of objects, creating a class-specific non-rigid registration model.

We show that incorporating this information not only increases the robustness of the registration for noisy and partially occluded objects, but also allows some ability to infer hidden or occluded parts. Rather than attempting to match the two point sets directly, our method tries to find a transformation linearly interpolated and extrapolated from other transformations found within the class which best matches the observed point set. This results in more likely shapes, as we restrict the transformations to be more closely resemble ones which we have actually observed during training. Fig. 1.1 demonstrates our method's ability to handle occlusions. Additionally, using these learned transformations, our method can be used to generate novel instances such as those in Fig. 1.2.

In the next section, we give a more concrete definition of the registration problem. In Chapter 2, we cover background work which serves as a basis and context for our own work as well as cover recent contributions which are similar to our work. In Chapter 3, we outline our method. In Chapter 4, we describe our experiments and present our results. And in Chapter 5, we provide our conclusions and present ideas for future work.

1.2. Problem Definition and Objectives

The point set registration problem is defined as: given two sets of points, find the underlying transformation which maps one point set to the second. More formally: given two sets of D-dimensional points, $Y = (y_1, ..., y_M)^T$ and $X = (x_1, ..., x_N)^T$, find a transformation T that when applied to the set Y, minimizes the distance between corresponding points in the set X. In general, a transformation T can be any function which takes in a set of N D-dimensional input points

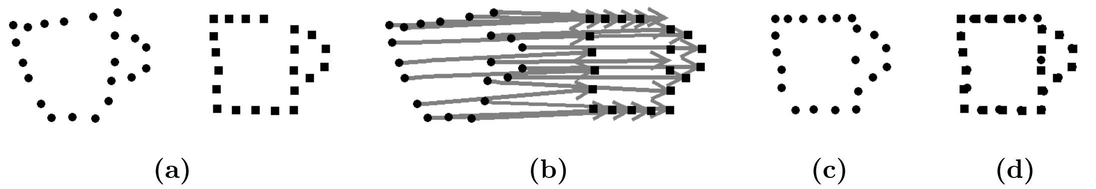


Figure 1.3.: A 2-D example of the registration problem. We seek to find the underlying transformation between the round and square points. (a) shows the two shapes, (b) shows a possible transformation linking the first shape to the second, (c) shows the result of the transformation, and (d) shows the transformed shape overlayed onto the reference shape.

and produces a set of N D-dimensional output points in the same order. The points are D-dimensional and need not represent only spacial locations, but can instead be higher dimensional feature vectors. The challenge is that not only is the true underlying transformation unknown, but so are the correspondences between points. Indeed, if we knew the correspondences, the transformation would be simple to estimate, and if we knew the transformation, we could easily estimate the correspondences. We must therefore estimate not only the transformation, but also the correspondences.

For any class of transformations which can be represented by a parameterization, each transformation can be thought of as a point in the space of all transformations. Within this space, for any set of transformations, there exists a linear subspace which approximately contains those transformations. During inference, rather than directly estimating the transformation between two shapes, our method searches this linear subspace for the transformation. Doing so restricts the transformations the shape can undergo to linear combinations of ones which were previously observed, adding robustness against noise and occlusion.

Our method is divided into two phases: a learning phase and an inference phase. In the learning phase, the objective is to create a class-specific linear model of the transformations a class of objects can undergo. We do this by first selecting a single model to be a canonical instance of the class, and then we find the transformations relating this instance to all other instances of the class using Coherent Point Drift (CPD) (Myronenko and Song, 2010). We then find a linear subspace of these transformations, which becomes our transformation model for the class. In the inference phase, the objective is, given a newly observed instance, search this subspace of transformations to find the transformation which best relates the canonical instance to the observed instance.

The transformations are represented by a single rigid transform, which accounts for small global misalignments, and a dense deformation field, which gives a transformation for every point in the space of the transforming shape. This allows points defined in the space of the canonical shape, such as manipulation information, to be transformed into the space of an observed instance. These points do not need to be known a priori and can be added at any time, even after the registration has completed. We also give an equation for estimating the inverse transformation, which allows applications such as taking manipulation knowledge gained in the space of the observed instances and transferring it back into the canonical space. This can then be used to aggregate and train on data collected after interactions with different instances. Additionally, the transformation model created in the training phase can be used to interpolate and extrapolate between instances to create novel instances of the class.

We summarize the objectives of our work:

1. Propose a learned non-rigid registration method which finds a low-dimensional space of deformation fields which can be used to interpolate and extrapolate between instances of a class of objects to create novel instances

2. and when given a coarsely-aligned partially- or fully-observed shape, finds a transformation which models the non-rigid deformation and alignment relating the shapes while being robust to noise and occlusion.

2. Related Work

2.1. Registration

Several solutions have been proposed for the problem of rigid registration. Chen and Medioni (1992) proposed the Iterate Closest Point algorithm, which alternates associating every point in one set with the closest point in the other set and then applying the rigid transformation which minimizes the distance between the corresponding points until a local minimum is reached. Since then, there have been numerous variants proposed (Rusinkiewicz and Levoy, 2001), including several variants which use non-rigid transformations (Brown and Rusinkiewicz, 2004, 2007; Haehnel et al., n.d.). These methods typically only find a local minimum, but for rigid transformations, recently Yang et al. (2016) proposed a method for finding the global optimum using branch-and-bound methods.

For non-rigid registration, we need to impose some a priori restrictions or regularization on the motion or deformation of the points between sets or else we have no way of establishing correspondences or estimating a transformation. Different transformation priors such as isometry (Bronstein et al., 2006; Ovsjanikov et al., 2010; Tevs et al., 2009), elasticity (Haehnel et al., n.d.), conformal maps (Kim et al., 2011; Lévy et al., 2002; Zeng et al., 2010), thin-plate splines (Allen et al., 2003; Brown and Rusinkiewicz, 2007), and Motion Coherence Theory (Myronenko and Song, 2010) have been used to allow for or to penalize different types of transformations. Additionally, the error function and correspondence search method have several variants as well (Rusinkiewicz and Levoy, 2001).

A significant portion of registration research is dedicated specifically to the human body. There exist many methods for non-rigid registration of human body parts such as hands (Oikonomidis et al., n.d.; Qian et al., 2014), faces (Blanz and Vetter, 1999; Bolkart and Wuhrer, 2015, 2016), and whole bodies (Allen et al., 2003; Hasler et al., 2009). In many of these cases, specialized shape or motion templates are learned or manually created.

Many techniques focus on general articulated shapes (Anguelov et al., 2004; Chang and Zwicker, 2009), focusing on matching the intrinsic shape of the object instead of or in addition to the extrinsic shape. Anguelov et al. (2004) use a joint probabilistic model over all point-to-point correspondences and the geodesic distance metric over the mesh between points to create a non-rigid registration method that works well even over large deformations of an articulated model without any prior shape or deformation dynamics knowledge. It functions even when one of the scans is only a partial view and can even be used to automatically recover the unknown underlying articulated skeleton. Another method focusing on articulated shapes is proposed by Chang and Zwicker (2009), who model an object's motion between scans as a reduced deformable model (RDM), where the object's motion is given by only a small set of parameters. The deformation model and surface representation are decoupled, with the articulation modeled by the RDM and the surface modeled by a linear blend skinning model. They show their method can deal with severe occlusions and missing data.

The medical field is a large contributor of non-rigid registration methods, particularly for the registration of images (Bernard et al., 2016; McInerney and Terzopoulos, 1996; Rohlfing and Maurer, 2003; Rohlfing, Maurer, et al., 2004; Rueckert, Frangi, et al., 2003; Rueckert, Sonoda, et al., 1999). Recently, Bernard et al. (2016) proposed using shape priors in the form of statistic shape models (SSM) to increase the robustness and accuracy of the registration and to deal with sparse data. Rohlfing and Maurer (2003) use parallelization to significantly speed up the nonrigid registration process, allowing it to be more widely applied to clinical situations where fast execution is required.

Many methods use non-rigid registration for surface reconstruction (Li, Adams, et al., 2009; Li, Sumner, et al., 2008; Newcombe et al., 2015; Süßmuth et al., 2008; Wand, Adams, et al., 2009; Wand, Jenke, et al., 2007). Methods such as Brown and Rusinkiewicz (2007) and Li, Sumner, et al. (2008) present surface reconstruction methods for global non-rigid registration. Methods such as Li, Adams, et al. (2009) and Zollhöfer et al. (2014) use a Kinect depth camera to capture an initially low-resolution 3D surface and use non-rigid registration to continuously add higher frequency details to the model with each new frame. Li, Vouga, et al. (2013) use non-rigid registration to create water-tight models of humans. The method is designed such that users can create 3D models of themselves by themselves, using non-rigid registration to account for the subtle changes in pose or surfaces such as clothing or hair while the user turns between scans.

In Newcombe et al. (2015), the authors propose a non-rigid dense surface reconstruction method using dense non-rigid registration. Using a depth camera to capture a sequence of depth images, they calculate a dense 6-dimensional warp field between frames. They then undo the transformations between each frame and then rigidly fuse the scans into one canonical shape using KinectFusion (Izadi et al., 2011). The dense warp field is estimated by minimizing an energy function combining a data term and a regularization term which penalizes non-smoothness and seeks to create a rigid-as-possible deformation.

Stueckler et al. (2011) develop an efficient real-time non-rigid registration method using Coherent Point Drift. They use a coarse-to-fine correspondence search and warp estimation of RBG-D images using their own feature representation called multi-resolution surfel maps (MRSMaps) (Stückler and Behnke, 2014), a multi-resolution representation which stores shape and color information in an octree.

2.2. Class-Level Shape Spaces

Several methods exist to create a parameterized space of shapes. Some of these methods use these shape spaces to add robustness or accuracy to previous methods or to provide additional versatility or features.

Blanz and Vetter (1999) create a morphable model of faces which can create novel faces and interpolate between faces using a few parameters. Additionally, it regularizes the naturalness of the modeled face and avoids faces it considers to be unlikely. Similarly, Allen et al. (2003) create a shape space of human bodies using human body range scans with sparse 3D markers. Hasler et al. (2009) extend this space to include pose, creating a unified space of both pose and body shape. This allows them to model the surface of a body in various articulated poses more accurately than previous methods by incorporating body shape information such as weight or muscularity.

Nguyen et al. (2011) attempt to optimize maps between shapes by creating collections of similar shapes and optimizing the mapping at a global scale. Huang et al. (2012) also use collections of shapes to enforce global consistency, but also creates a small collection of base shapes from which correspondences are established between all other shapes.

Burghard et al. (2013) propose an approach to estimate dense correspondences on a shape given initial coarse correspondences. They use the idea of minimum description length to create a compact shape space of related shapes of strongly varying geometry. By minimizing the the description length of the shape space, they avoid overfitting and make the model more statistically meaningful. Additionally, they consider a shape to be an assembly of deformable, dockable parts, learning an individual shape space for each part, and then seemlessly and continuously merge the parts into a single mesh.

Engelmann et al. (2016) use 3D shape priors as regularization cues to segment objects and estimate their shape and pose in stereo images. They learn a compact shape manifold which represents a class's intra-class shape variance, allowing them to infer shape in occluded regions or regions where data might be missing or noisy such as textureless, reflective, or transparent surfaces. They demonstrate that these shape priors improve scene reconstruction and pose estimation in stereo images.

2.3. Discussion

Current state-of-the-art non-rigid registration methods give good results but have limitations. Newcombe et al. (2015) are able to deform a scene in real-time toward a canonical shape, but use optical flow constraints and thus are not suited for large deformations or strong changes in illumination and color. Many of the methods are capable of handling large deformations, but not occlusions. For example, the method proposed by Burghard et al. (2013) accurately estimates dense correspondences, but does not perform well in the presence of noise or incomplete scans, such as from occlusions. Methods such as Engelmann et al. (2016) can handle large occlusions and minor misalignments to uncover the shape and alignment of the object, but do not give correspondences between points and do not offer any kind of transformation.

We propose a new method for non-rigid registration, which incorporates class-level information. By incorporating this information into the registration, we can avoid unlikely shapes and focus on deformations actually observed in the class. Furthermore, the space can be used to create novel instances and interpolate and extrapolate between previous ones. In the next section, we give a detailed overview of our method.

3. Method

3.1. Overview

In this section, we describe our non-rigid registration method. Our method is a learned method, and thus has an initial training phase in which the transformation model for a class of objects is built. Once a transformation model is built, inferencing can be performed to find a transformation which relates one of the shapes from the class called the canonical shape to fully- or partially-observed novel instances. The transformation is given as a single rigid transformation plus a dense deformation field. Sections 3.2, 3.3, and 3.4 explain the steps to building a trained model, while Section 3.5 explains how to use the model to find transformations relating the canonical shape and any additional points defined in its space to an observed shape. Section 3.6 shows how to estimate the inverse deformation, allowing points in the space of the observed shape to be transformed back into the space of the canonical shape. Finally in Section 3.7, we present a few variants to our method. Summaries of the training and inference steps are given by Alg. 1 and Alg. 2 respectively.

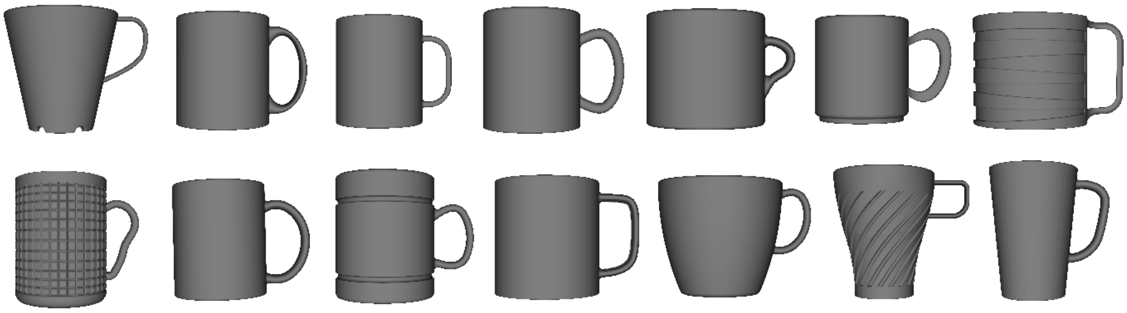


Figure 3.1.: Several instances belonging to the class *Mug*. All the examples are in the class's canonical pose: standing upright with the handle to the right.

3.2. Classes and Shape Representation

In order to make use of class-level information, we must first define the notion of a class. We define a class as a set of objects which share the same topology and a similar extrinsic shape. Figure 3.1 demonstrates several instances belonging to the class *Mug*. Each class of objects is composed a set of I training example shapes $\mathbf{E}$. To represent these shapes, we use a point cloud with each point in $\mathbb{R}^3$. The points are downsampled uniformly-sampled locations on the training objects' surfaces. For the purposes of equations, an N-point point cloud is considered an $N \times 3$ matrix.

If starting with a mesh, a point cloud can be generated by ray-casting. Several viewpoints are selected on a tesselated sphere and rays are cast to the surface of the object. When a ray collides with the surface, a point is added to a point cloud for that view. The point clouds generated from each view are then rotated into a single canonical view and then merged. Finally, a downsampling step is performed with a voxel grid by merging points which fall into the same voxel. For each voxel containing points, the points are replaced with a single point whose location is the average of those points within the same voxel. To maintain information such as color or vertex connectivity, the original mesh can be kept and then later transformed using the transformations given by the transformation model just the same as any other information defined in the same space. However, information such as surface normals, which depend on the then deformed surface, will need to be recomputed.

Each class additionally specifies a single instance $\mathbf{C} \in \mathbf{E}$ to represent the canonical shape of that class. This can be chosen by heuristics or chosen as the shape with the lowest reconstruction energy after finding the deformation field between all other $I-1$ training examples:

$$\mathbf{C} = \mathrm{argmin}_{\mathbf{E}^j} \frac{-1}{(I-1)} \sum_{i=1}^{I} (1-\delta_{ij}) \log \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \exp(\frac{-(\|D_i(\mathbf{E}^j_m) - \mathbf{E}^i_n\|)^2}{2\sigma}) \tag{3.1}$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i == j \\ 0 & \text{otherwise} \end{cases}$$

$\mathbf{E}^j$ is the jth training example, $\mathbf{E}^j_m$ and $\mathbf{E}^i_n$ are the mth and nth points of the jth and ith training examples, respectively; and $D_i(\mathbf{E}^j_m)$ represents the deformation field associated with the ith training example applied to the mth point of the

j training example. σ can be set to any value and exists mostly to increase the numerical stability of the equation. The process of obtaining the deformation fields $D_i(\cdot)$ is explained in the next section. The remaining shapes are taken as training instances $\mathbf{T} = \{\mathbf{E} - \mathbf{C}\}$.

Additionally, each class must specify a canonical pose and reference frame. The pose is important for initial alignment and to remove spurious transformations, and the reference frame is important for correlating transformations and dimension reduction. For example: for mugs, a sensible pose is one where the top is open upward with all the handles in the same position, and a sensible reference frame may be at the bottom-center of the mug's cylinder; while for chairs, the best reference frame may be one centered on the seat or at the center of the base of the legs.

3.3. Coherent Point Drift

Once we have a set of training examples and a canonical shape, we need to find transformations which map the canonical shape to all other training shapes. Here we make use of another non-rigid registration method called Coherent Point Drift (CPD) (Myronenko and Song, 2010). CPD imposes a smoothness constraint on the deformation of the points in the form of motion coherence, which is based on Motion Coherence Theory (Yuille and Grzywacz, 1988). The idea is that points near each other should move coherently and have a similar motion to their neighbors. Rather than a hard assignment of points, such as in ICP (Chen and Medioni, 1992), points are soft-assigned to all other points based on their proximity. The motion of the point is determined by these soft-assignments and thus points tend to move like their neighbors.

We choose CPD here for several reasons. CPD provides a dense deformation field, allowing us to find deformation vectors for novel points, even those added after the field is created, which in turn allows us to apply the method to manipulation skill transfer as well as several other applications. Additionally, CPD allows us to create a feature vector representing the deformation field which is of constant length across training examples and where elements in the vector correspond with the same elements in another. This allows us to apply Principle Component Analysis (PCA) to these feature vectors in order to find a lower-dimensional deformation field manifold, which is explained in the next section.

Given two point sets, a template set $\mathbf{S}^{[t]} = (s_1^{[t]}, ..., s_M^{[t]})^T$ and a reference point set $\mathbf{S}^{[r]} = (s_1^{[r]}, ..., s_N^{[r]})^T$, CPD tries to estimate a deformation field mapping the points in $\mathbf{S}^{[t]}$ to $\mathbf{S}^{[r]}$. The points in $\mathbf{S}^{[t]}$ are considered centroids of a Gaussian

Mixture Model (GMM) from which the points in $\mathbf{S}^{[r]}$ are drawn. CPD seeks to maximize the likelihood of the GMM while imposing limits on the motion of the centroids. Once the field is estimated, a function v gives a deformation vector for every point in $\mathbf{S}^{[t]}$ such that:

$$\mathbf{S}^{[t^*]} = v(\mathbf{S}^{[t]}) + \mathbf{S}^{[t]} \tag{3.2}$$

v is defined for any set of N D-dimensional points $\mathbf{Z}$ as:

$$v(\mathbf{Z}) = G(\mathbf{S^{[t]}}, \mathbf{Z})\mathbf{W} \tag{3.3}$$

where $G_{N\times M}$ is a Gaussian kernel which is defined element-wise as:

$$g_{ij}(\mathbf{S}^{[t]}, \mathbf{Z}) = \exp\bigl(-\frac{1}{2}\left\|\frac{\mathbf{Z}_i - \mathbf{S}_j^{[t]}}{\beta}\right\|^2) \tag{3.4}$$

$\mathbf{Z}_i$ and $\mathbf{S}_j^{[t]}$ represent the ith and jth rows of $\mathbf{Z}$ and $\mathbf{S^{[t]}}$ respectively, and $\mathbf{W}_{M\times D}$ is a matrix of kernel weights. An additional interpretation of $\mathbf{W}$ is as a set of D-dimensional deformation vectors, each associated with one of the M points of $\mathbf{S^{[t]}}$. As a point $z \in \mathbf{Z}$ grows nearer to a point $s_j^{[t]} \in \mathbf{S^{[t]}}$, the more the vector $w_j \in \mathbf{W}$ associated with point $s_j^{[t]}$ will influence the deformation of z. The parameter β controls the strength of interaction between points. Note that N can be any value while D is determined by the field. If $\mathbf{Z} = \mathbf{S^{[t]}}$, then Eq. 3.2 gives the transformation relating $\mathbf{S^{[t]}}$ to $\mathbf{S^{[r]}}$.

The results of $G(\cdot,\cdot)$ can simply be computed by Eq. 3.4, but the matrix $\mathbf{W}$ needs to be estimated. CPD uses an Expectation Maximization (EM) algorithm derived from the one used in Gaussian Mixture Models (Bishop, 1995) to find this matrix.

The energy function we want to minimize is:

$$E(\mathbf{W}) = -\sum_{n=1}^{N}\log\sum_{m=1}^{M}\exp(-\frac{1}{2}\left\|\frac{s_n^{[r]} - s_m^{[t]} - v(s_m^{[t]})}{\sigma}\right\|^2) + \frac{\lambda}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{G}\mathbf{W}) \tag{3.5}$$

where here the matrix $\mathbf{G}$ is defined as:

$$\mathbf{G} = G(\mathbf{S}^{[t]}, \mathbf{S}^{[t]})$$

The first term penalizes distance between points after applying the transformation, and the second term $\frac{\lambda}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{G}\mathbf{W})$ is a regularization term which enforces motion coherence. The parameter λ controls the weight between the two penalties.

The parameter σ controls the range of all Gaussian mixture components, with a smaller σ indicating a more localized range for each Gaussian.

In our method, the parameter σ is given by the equation:

$$\sigma = \left| M * \mathrm{tr}(\mathbf{S}^{[r]T}\mathbf{S}^{[r]}) + N * \mathrm{tr}(\mathbf{S}^{[t]T}\mathbf{S}^{[t]}) - \frac{2 * \mathrm{sum}(\mathbf{S}^{[r]}) * \mathrm{sum}(\mathbf{S}^{[t]T})}{M * N * D} \right|$$

where the function sum$(\cdot)$ gives the sum of each column, M is the number of points in $\mathbf{S}^{[t]}$, N is the number of points in $\mathbf{S}^{[r]}$, and D is the dimensionality of the points.

From Eq. 3.5, the authors derive an upper bound function:

$$Q(\mathbf{W}) = \sum_{n=1}^{N}\sum_{m=1}^{M} \mathbf{P} \frac{\left\| \mathbf{S}_n^{[r]} - \mathbf{S}_m^{[t]} - v(\mathbf{S}_m^{[t]}) \right\|^2}{2\sigma^2} + \frac{\lambda}{2}\mathrm{tr}(\mathbf{W}^T\mathbf{G}\mathbf{W}) \tag{3.6}$$

The upper bound is reduced by alternating between an E- and M-step until some convergence criteria is met.

In the E-step, the posterior probabilities matrix $\mathbf{P}$ is estimated using the previous parameter values. To add robustness to outliers, an additional uniform probability distribution function component is added to the mixture model. The matrix $\mathbf{P}$ is defined element-wise as:

$$p_{mn} = e^{-\frac{1}{2}\left\| \frac{\mathbf{s}_m^{[t]} - \mathbf{s}_n^{[r]}}{\sigma} \right\|^2} \Big/ \left(\frac{(2\pi\sigma^2)^{\frac{D}{2}}}{a} + \sum_{m=1}^{M} e^{-\frac{1}{2}\left\| \frac{\mathbf{s}_m^{[t]} - \mathbf{s}_n^{[r]}}{\sigma} \right\|^2} \right) \tag{3.7}$$

where a defines the support of the uniform pdf.

In the M-Step, the matrix $\mathbf{W}$ is estimated by solving the equation:

$$(\mathrm{diag}(\mathbf{P}\overline{1})\mathbf{G} + \lambda\sigma^2\mathbf{I})\mathbf{W} = \mathbf{P}\mathbf{S}^{[r]} - \mathrm{diag}(\mathbf{P}\overline{1})\mathbf{S}^{[t]} \tag{3.8}$$

where $\overline{1}$ represents a column vector of M ones, $\mathbf{I}_{M\times M}$ is the identity matrix, and the function diag$(\cdot)$ creates a diagonal matrix.

After each step, deterministic annealing is used to reduce σ in order to simulate a coarse-to-fine matching strategy:

$$\sigma = \alpha\sigma$$

The parameter α controls the annealing rate. The authors recommend a value between $[0.92, 0.98]$.

In our method, we use the canonical shape $\mathbf{C}$ for the deforming template shape

$\mathbf{S}^{[\mathbf{t}]}$ and each training example $\mathbf{T}_i$ as the reference point set $\mathbf{S}^{[\mathbf{r}]}$. $D_i(\cdot)$ is therefore defined as

$$D_i(\mathbf{C}) = v_i(\mathbf{C}) = \mathbf{G}\mathbf{W}_i$$

where $\mathbf{W}_i$ is the $\mathbf{W}$ matrix computed by taking training example $\mathbf{T}_i$ as the reference point set $\mathbf{S}^{[\mathbf{r}]}$.

3.4. Low-Dimensional Deformation Field Manifold

After finding a deformation field which relates the canonical model to all other training examples, we need a feature vector to represent each deformation field so that we can find a manifold containing them. From Eq. 3.3, we see that the deformation field function is defined by matrices $\mathbf{G}$ and $\mathbf{W}$. $\mathbf{G}$ is defined by $G(\mathbf{C}, \mathbf{C})$ and therefore is only in terms of the canonical shape and remains constant for all training examples. The entire uniqueness of the deformation field for each training example is captured by the matrix $\mathbf{W}$.

For each training example $\mathbf{T}_i$, we take the matrix $\mathbf{W}_i$ from its deformation field and convert it into a row vector. This becomes the feature descriptor of that deformation field. The vectors are then assembled into a design matrix $\mathbf{Y}$. Finally, we apply principle component analysis (PCA) on this matrix to find the lower-dimensional manifold of deformation fields for this class.

PCA finds a matrix $\mathbf{L}_{p \times q}$ of principle components that can be used to estimate a matrix of n observation vectors $\mathbf{Y}$ of p features given a small set of q latent variables such that $q < n$ and $q \ll p$:

$$\mathbf{Y} = \mathbf{X}\mathbf{L}^T + \mathbf{Y}_\mu \tag{3.9}$$

where $\mathbf{Y}_\mu$ is a matrix composed of the mean vector of the training observations $\bar{Y}_\mu$ repeated row-wise for the n observation vectors. A lower value of q increases the generalization capacity at the cost of attributing more of the variance as noise and discarding it.

The matrix $\mathbf{Y}$ can also be converted from observed space into latent space by the equation:

$$\mathbf{X} = (\mathbf{Y} - \mathbf{Y}_\mu)\mathbf{L} \tag{3.10}$$

We find the matrix L using the PCA Expectation Maximization (EM) algorithm (Roweis, n.d.). We choose this method over more analytical algorithms due to its superior performance in situations with high dimensions and scarce data and also because of its greater numerical stability.

Much like with CPD, we alternate between an E- and M-step. The E-Step is given by the equation:

$$\mathbf{X} = \mathbf{Y}_c\mathbf{L}^T(\mathbf{L}\mathbf{L}^T)^{-1} \tag{3.11}$$

and the M-Step is given by the equation:

$$\mathbf{L} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}_c \tag{3.12}$$

where $\mathbf{Y}_c = \mathbf{Y} - \mathbf{Y}_\mu$, i.e. the mean-centered observations.

The method is shown to eventually converge to a local minimum using standard EM convergence proofs (Dempster et al., 1977). Additionally, it has been shown that the only stable local extremum is the global maximum (Tipping and Bishop, 1999a,b), meaning the algorithm will always converge to the correct result with enough iterations.

We use the design matrix created from the set of $\mathbf{W}$s as vectors as our observed data $\mathbf{Y}$ and apply PCA to estimate the deformation field manifold for the class of objects, represented by the matrix $\mathbf{L}$ and the mean observation vector $\bar{Y}_\mu$.

Using Eq. 3.9 and Eq. 3.10, a deformation field can now be described by only q latent parameters. Any parameters of a transformation $\mathbf{W}$ can now be approximated as a q-dimensional point in space by first converting it into a vector $\bar{Y}$ and then applying Eq. 3.10. And similarly, any point $\bar{X}$ in the space can be converted into the parameters of a transformation by first applying Eq. 3.9 and then converting it into a $M \times 3$ matrix $\mathbf{W}$. Moving through the q-dimensional space linearly interpolates between the transformations.

The matrix $\mathbf{L}$, vector $\bar{Y}_\mu$, and canonical shape $\mathbf{C}$ together represent the transformation model for a class. Alg. 1 gives a summary of the training steps to build the transformation model.

3.5. Inference

Once we have built a transformation model with the matrix $\mathbf{L}$, vector $\bar{Y}_\mu$, and the canonical shape $\mathbf{C}$, we can begin to register the canonical shape to novel instances and estimate the underlying transformation. The parameters of the transformation are given by the q parameters of the latent vector $\bar{X}$ plus an additional 7 parameters of a rigid transformation θ, represented by a position vector and an axis-angle. The rigid transformation accounts for minor misalignments in position and rotation between the target shape and the canonical shape at the global level. The method

```
Algorithm 1: Building the Transformation Model for a Class
  Input: A set of training shapes $\mathbf{E}$ in their canonical pose and reference frame
    1. Select a canonical shape $\mathbf{C}$ via heuristic or Eq. 3.1
    2. Estimate the deformation fields between the canonical shape and the other training examples using CPD
    3. Concatenate the resulting set of $\mathbf{W}$ matrices from the deformation fields into a design matrix $\mathbf{Y}$
    4. Perform PCA on the design matrix $\mathbf{Y}$ to compute the latent space of deformation fields
  Output: A canonical shape $\mathbf{C}$ and a latent space of deformation fields represented by $\mathbf{L}$ and $\bar{Y}_\mu$
```

does not model the rotation of points, such as grasp poses, but these rotations can be estimated. Section C details this process.

For a coarsely-aligned partially- or fully-observed shape $\mathbf{O}$, represented by a 3-dimensional point cloud, we want to find the underlying transformation relating the canonical model to the shape. We concurrently optimize for shape and pose using gradient descent. We expect many local minimum, especially with regard to pose, and therefore require an initial coarse alignment of the observed shape. Using the processes described in previous sections, the q parameters of $\bar{X}$ represent the parameters $\mathbf{W}$ of a dense deformation field. The global pose is represented by the 7 rigid transformation parameters, 3 for position and 4 for the rotation. In total, there are $q + 7$ parameters to optimize. Because we cannot estimate the gradient analytically, we instead compute it numerically.

We want to find an aligned dense deformation field, which when applied to the canonical shape $\mathbf{C}$, minimizes the distance between corresponding points in observed shape $\mathbf{O}$. Specifically we want to minimize the energy function:

$$E(\bar{X}, \theta) = -\log \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \exp\left(\frac{\|(\mathbf{D}_m - \mathbf{O}_n)\|^2}{2\sigma}\right) \tag{3.13}$$

$\mathbf{D}$ is the transformed canonical shape given by the equations $\mathbf{D} = \Theta_\theta[\mathbf{C} + v(\mathbf{C})]$, where the function v is the deformation field created from $\bar{X}$, and the function Θ applies the rigid transformation θ by rotating the points via the Rodriquez formula (Sec. A) and then adding the translation vector.

When a transformation is found that minimizes the energy, we can transform any point or set of points into the observed space by calculating and applying the deformation vector using Eq. 3.2 and applying the rigid transformation. This includes the canonical shape itself, which can represented as the point cloud used to find the transformation or as a mesh or any other representation. Alg. 2 summarizes the inference process. Step 2.a.iv. is for certain variants presented in a later section which use surface normals for correspondence rejection and is unnecessary for the base method presented here.

```
Algorithm 2: Inference with the Transformation Model
  Input: Transformation model ($\mathbf{C}$, $\mathbf{L}$, $\bar{Y}_\mu$) and observed shape $\mathbf{O}$
    1. Compute matrix $\mathbf{G} = G(\mathbf{C}, \mathbf{C})$ (Eq. 3.4)
    2. Use gradient descent to estimate the parameters of the underlying
       transformation ($\bar{X}$ and $\theta$) until the termination criteria is met:
        a) Using the current values of $\bar{X}$ and $\theta$:
             i. Use Eq. 3.9 to create vector $\bar{Y}$ and convert it into matrix $\mathbf{W}$
            ii. Use Eq. 3.3 and Eq. 3.2 to deform $\mathbf{C}$
           iii. Apply the rigid transformation to the deformed $\mathbf{C}$
            iv. (Estimate the normals of the transformed $\mathbf{C}$)
             v. Compute the energy using Eq. 3.13
  Output: Non-rigid transformation given by deformation field description
  $\mathbf{W}$ and rigid transform $\theta$
```

3.6. Inverse Deformation

The inverse of the deformation v^{-1} is not directly available, but it can be approximated for a point z in the space of the observed shape using a set of points $\mathbf{Y} = (y_1, ..., y_M)^T$ in the canonical space which deform close to z with the equation:

$$v^{-1}(z) = -\frac{\sum_{i=1}^{M} G(z, y_i + v(y_i))v(y_i)}{\sum_{i=1}^{M} G(z, y_i + v(y_i))} \tag{3.14}$$

3.7. Variants

We also developed several variants to our inferencing method. The correspondences between points is critical for estimating a transformation, so we explored a few variations which use correspondence rejection. We also want to find the transformation between the surfaces of the objects, not necessarily the points representing those surfaces, and thus explored a variant that used the point-plane distance error metric rather than the simpler point-point distance error metric. And finally, we propose variants which use a robust loss function, which attempts to reduce the affect of outliers.

3.7.1. Normals Rejection

In many point-set registration methods, it is common to reject correspondences based on certain criteria. One of the common criteria is normal compatibility. By changing our energy function to the equation:

$$E(\bar{X}, \theta) = -\log \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \exp\left(\frac{-\|\mathbf{D}_m - \mathbf{O}_n\|^2}{2\sigma}\right) * \Delta(\hat{n}_m^{\mathbf{D}} \cdot \hat{n}_n^{\mathbf{O}}) \tag{3.15}$$

where

$$\Delta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

we use the dot product between the normals of each surface between each point in order to reject points which lie on opposite surfaces. This is to avoid correspondences with surfaces which may not actually be associated, e.g. the inner and outer walls of a mug. This is particularly a problem in the case of partial views, where a corresponding surface may be occluded, causing ambiguity with the surfaces of the canonical shape.

If the canonical shape is generated from a mesh, an initial set of normals for the shape can be given by assigning each point the surface normal from the surface it was generated from. We re-estimate the normals of the deformed canonical shape each step using SVD B. Due to the ambiguity in sign, we flip a normal n_i at step t if is more than 90° away from its previous estimation at step $t-1$:

$$\hat{n}_i^{[t]} = \begin{cases} \hat{n}_i^{[t]} & \text{if } \hat{n}_i^{[t]} \cdot \hat{n}_i^{[t-1]} > 0 \\ -\hat{n}_i^{[t]} & \text{otherwise} \end{cases}$$

If $\mathbf{O}$ is given without normals, we can estimate them using this technique as well, but with replacing the conditional with $\hat{n}^{[t]} \cdot (v_p - p_i) > 0$ where v_p is the viewpoint and p_i is the point the normal corresponds to.

3.7.2. Normals Rejection Plus Point-Plane Distance Metric

Another variant, based on the one presented in the previous section, replaces the point-point distance error metric with a point-plane distance metric. The energy function is given by the equation:

$$E(\bar{X}, \theta) = -\log \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \exp\left(\frac{-\left((\mathbf{D}_m - \mathbf{O}_n) \cdot \hat{n}_n^{\mathbf{O}}\right)^2}{2\sigma} \right) * \Delta(\hat{n}_m^{\mathbf{D}} \cdot \hat{n}_n^{\mathbf{O}}) \quad (3.16)$$

where again

$$\Delta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

and $\hat{n}_m^{\mathbf{D}}$ and $\hat{n}_n^{\mathbf{O}}$ represents the surface normal associated with the mth and nth point of the deformed and observed shapes, respectively. The idea is to attempt to match the surfaces of two shapes rather than the points representing these surfaces. The surface is considered tangent to the normal, and thus minimizing this distance minimizes the distance to the plane formed by the point and its normal.

3.7.3. Robust Loss Function

A final variant was to place a robust loss function $\rho(\cdot)$ around the error of each point. For the original method, the robust energy function is given as:

$$E(\bar{X}, \theta) = -\log \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \rho \left(\exp(\frac{\|(\mathbf{D}_m - \mathbf{O}_n)\|^2}{2\sigma}) \right) \quad (3.17)$$

Similarly, for the normals rejection method:

$$E(\bar{X}, \theta) = -\log \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \rho \left(\exp\left(\frac{-\|\mathbf{D}_m - \mathbf{O}_n\|^2}{2\sigma} \right) * \Delta(\hat{n}_m^{\mathbf{D}} \cdot \hat{n}_n^{\mathbf{O}}) \right) \quad (3.18)$$

And finally, for the point-plane method:

$$E(\bar{X}, \theta) = -\log \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \rho \left(\exp\left(\frac{-\left((\mathbf{D}_m - \mathbf{O}_n) \cdot \hat{n}_n^{\mathbf{O}}\right)^2}{2\sigma} \right) * \Delta(\hat{n}_m^{\mathbf{D}} \cdot \hat{n}_n^{\mathbf{O}}) \right) \quad (3.19)$$

$\rho(\cdot)$ is a robust loss function chosen to minimize the effect of outliers, which can appear in noisy scans or when the object is not perfectly segmented from the scene. Specifically, we use the Huber Loss function (Huber et al., 1964):

$$\rho(a) = \begin{cases} \frac{1}{2}a^2 & \text{if } |a| \leq \epsilon \\ \epsilon(|a| - \frac{1}{2}\epsilon) & \text{otherwise} \end{cases} \tag{3.20}$$

where ϵ controls the sensitivity to larger values.

4. Experiments

4.1. Experimental Setup

In this section, we explain our experimental setup and present the results of our non-rigid registration technique. We tested our method on a set of meshes combined from a subset of the SHREC 2007 dataset (Jayanti et al., 2006) and a free online CAD database, GrabCad (*GrabCad* 2017). The meshes were converted into point clouds as detailed in the previous section. In total, we had 4 classes of objects: mugs, fish, four-legged animals, and airplanes. Samples (as meshes) from each class are shown in Fig. 4.1.

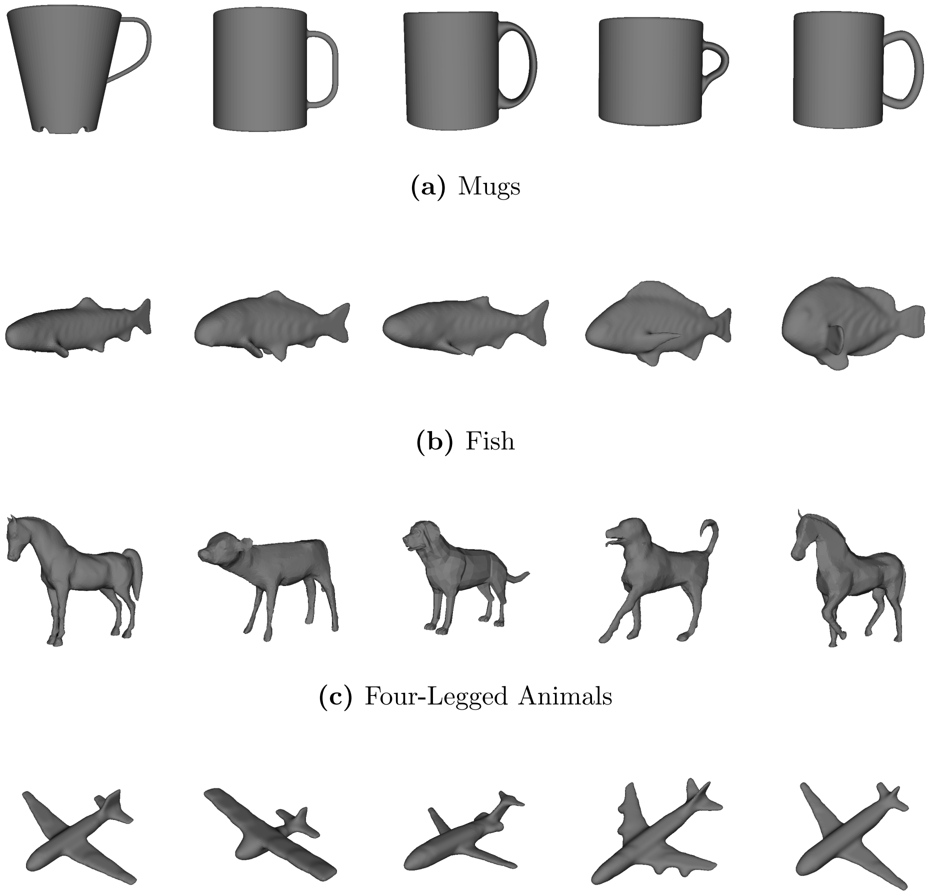
We tested our method's robustness to noise, misalignment, and occlusion and compared our results against results given by CPD. We tested with various levels of noise and misalignment. Each test was run on a full view of the object and on ten different partial views of the object. To obtain partial views, we used ray-casting using the same process as in the earlier section but from only one viewpoint.

Noise was simulated by randomly sampling a point from a normal distribution and scaling it by the noise factor and then adding that value to one coordinate of a point for every coordinate of every point. Misalignment was generated by uniformly sampling 7 values, scaling them by the misalignment factor, creating a rigid transformation from these parameters, and then applying it to the observed shape.

We then registered the canonical shape to the either misaligned or noisy partially- or fully-observed shape. We are interested in the methods' ability to find a transformation which matches the shape *without* occlusions or noise, and so we take the noiseless fully-observed shape as the ground truth. The misalignment is applied to both the observed and ground truth shapes.

In all experiments, we used the parameters $a = 1, \alpha = 0.98, \beta = 1$, and $\lambda = 3$ for training, and $a = 0.1$, $\sigma = 0.05$, and $\epsilon = 1.0$ for inferencing. q was set to capture at least 95% of the variance of each class, and each class's canonical shape was the one that gave the least reconstruction error (i.e. selected by Eq. 3.1).

We used two error functions to evaluate each method, an average minimum



(a) Mugs

(b) Fish

(c) Four-Legged Animals

(d) Airplanes

Figure 4.1.: Several samples used during our experiments.

		Base	**Noise**					**Misalignment**				
		0.0	**0.01**	**0.02**	**0.03**	**0.04**	**0.1**	**0.1**	**0.2**	**0.3**	**0.4**	**1.0**
Full	**CPD**	**0.035**	**0.036**	**0.037**	**0.038**	**0.039**	**0.051**	**0.035**	**0.036**	**0.036**	**0.037**	**0.040**
	Us	0.068	0.065	0.065	0.067	0.071	0.128	0.071	0.097	0.149	0.340	0.424
Part	**CPD**	0.121	0.122	0.121	0.120	0.118	0.111	0.121	0.122	**0.122**	**0.123**	**0.128**
	Us	**0.079**	**0.079**	**0.078**	**0.082**	**0.084**	**0.109**	**0.081**	**0.104**	0.182	0.177	0.433

Table 4.1.: Point error metric

		Base	**Noise**					**Misalignment**				
		0.0	**0.01**	**0.02**	**0.03**	**0.04**	**0.1**	**0.1**	**0.2**	**0.3**	**0.4**	**1.0**
Full	**CPD**	**0.003**	**0.003**	**0.002**	**0.002**	**0.002**	**0.001**	**0.003**	**0.003**	**0.003**	**0.003**	**0.004**
	Us	0.006	0.006	0.004	0.005	0.005	0.014	0.005	0.008	0.014	0.021	0.086
Part	**CPD**	0.017	0.017	0.016	0.015	0.015	0.009	0.016	0.017	0.017	0.017	**0.019**
	Us	**0.006**	**0.006**	**0.006**	**0.006**	**0.006**	**0.006**	**0.006**	**0.008**	**0.012**	**0.014**	0.059

Table 4.2.: Plane error metric

point distance error:

$$E_{\text{point}}(D, O^*) = \frac{1}{N} \sum_{n=1}^{N} \text{min}_m \left(\|D_m - O_n^*\|\right) \tag{4.1}$$

and an average minimum plane distance error:

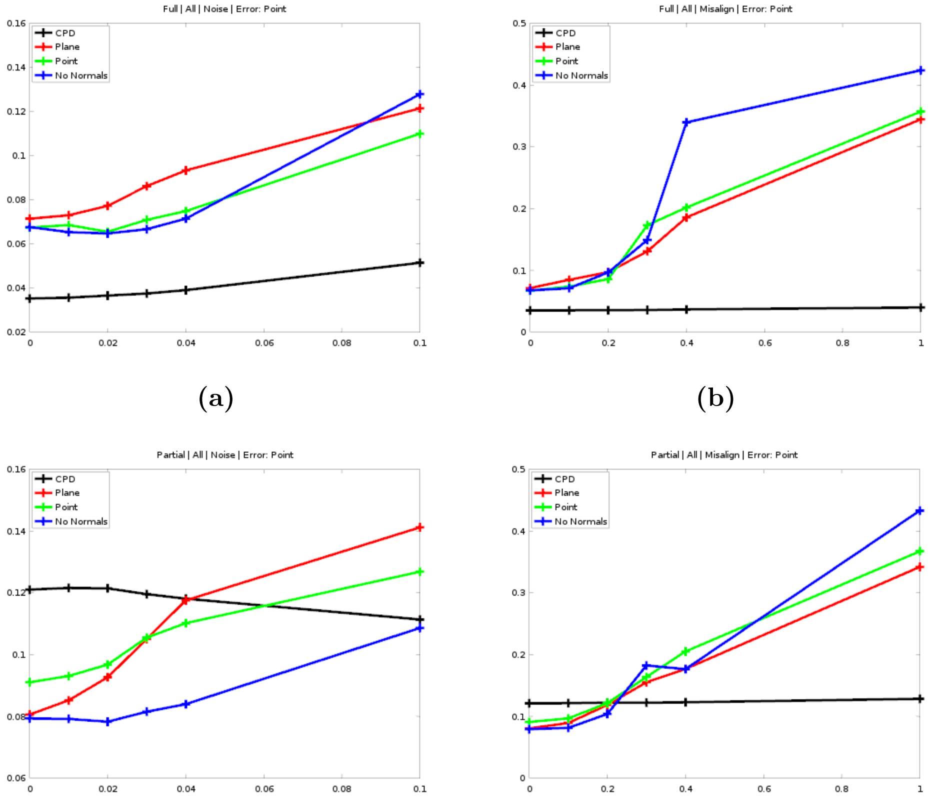
$$E_{\text{plane}}(D, O^*) = \frac{1}{N} \sum_{n=1}^{N} \text{min}_m \left((D_m - O_n^*) \cdot \hat{n}_n^{[O^*]}\right) \tag{4.2}$$

where D is the transformed canonical shape, O^* is the ground truth shape, and $\hat{n}_n^{[O^*]}$ is the normal of nth point of O^*. The results of the experiments are given in the next section.

4.2. Results

The results of our experiments are given in Tab. 4.1 and Tab. 4.2 and plotted in Fig. 4.2 and Fig. 4.3. When a shape is fully visible and without any noise, CPD outperforms our method. This is rather to be expected as CPD is the source of training data from which we build the registration model. Since our method searches a subspace created from the results of CPD, these results form an upper limit on performance in this scenario. However, when the shape becomes partially occluded, the method begins to outperform CPD, maintaining an error rate similar to when the entire shape is visible. We show a few examples of our method's results on partially-occluded shapes versus the results of CPD in Fig. 4.6.

We find that the variants generally perform quite similar to one another, with the no normal information version, i.e. the base version, slightly outperforming



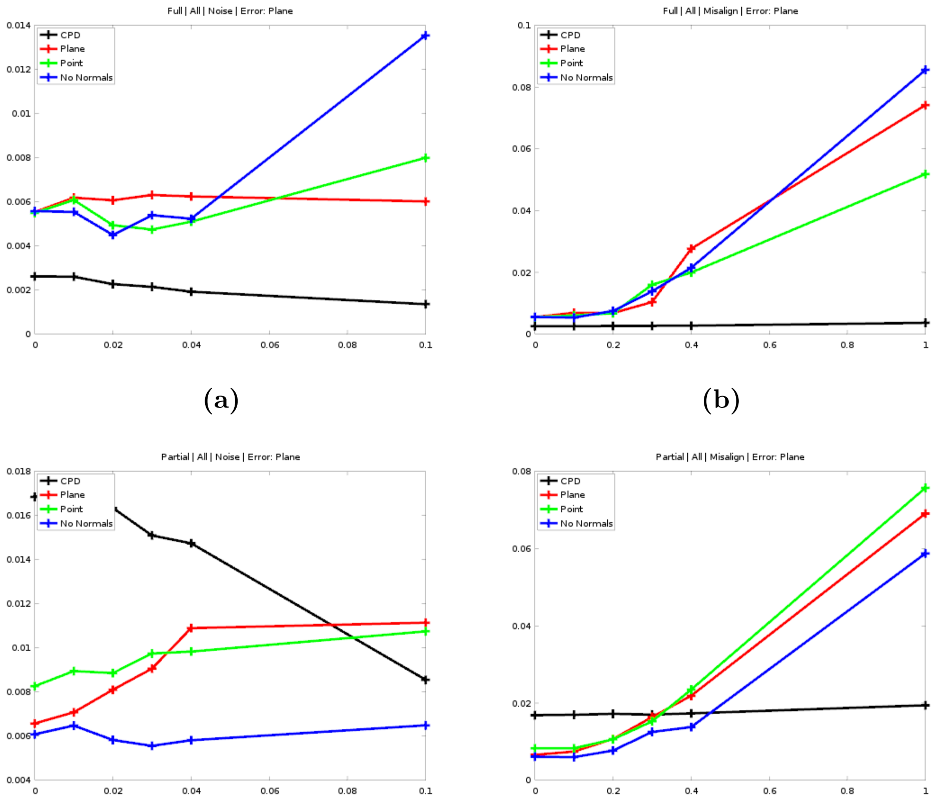
(a)

(b)

(c)

(d)

Figure 4.2.: Plot of the average point-point error for various noises and misalignments. (a) and (b) represent the average error on fully visible shapes with various levels of noise and misalignment respectively; while (c) and (d) give the average error on partially occluded shapes, again for noise and misalignment respectively. The red *Plane* method is the variant using the point-plane distance metric and normal rejection, the green *Point* method is the variant using the point-point distance metric and normal rejection, and the blue *No Normals* method is the base method using the point-point distance and no normal rejection.



(a)

(b)

(c)

(d)

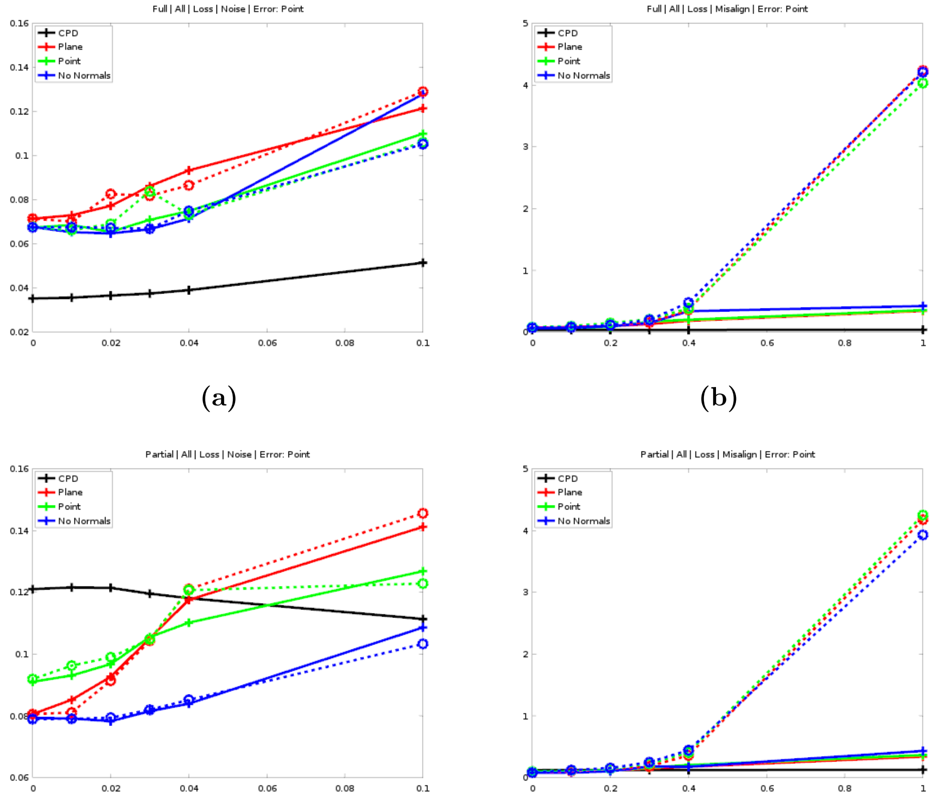
Figure 4.3.: The same plots as in Fig. 4.2 but with the average point-plane error instead.

		Mugs	Fish	Animals	Planes
Full	**CPD**	**0.0293**	**0.0307**	**0.0431**	**0.0379**
	Us	0.0521	0.0354	0.0942	0.0887
Part	**CPD**	0.1401	0.0981	0.1334	0.1125
	Us	**0.0570**	**0.0430**	**0.1053**	**0.1120**

Table 4.3.: Point error metric results by class with zero noise or misalignment

		Mugs	Fish	Animals	Planes
Full	**CPD**	**0.0030**	0.0024	**0.0024**	**0.0027**
	Us	0.0112	**0.0022**	0.0062	0.0027
Part	**CPD**	0.0286	0.0160	0.0137	0.0090
	Us	**0.0109**	**0.0029**	**0.0067**	**0.0038**

Table 4.4.: Plane error metric results by class with zero noise or misalignment



Full | All | Loss | Noise | Error: Point

Full | All | Loss | Misalign | Error: Point

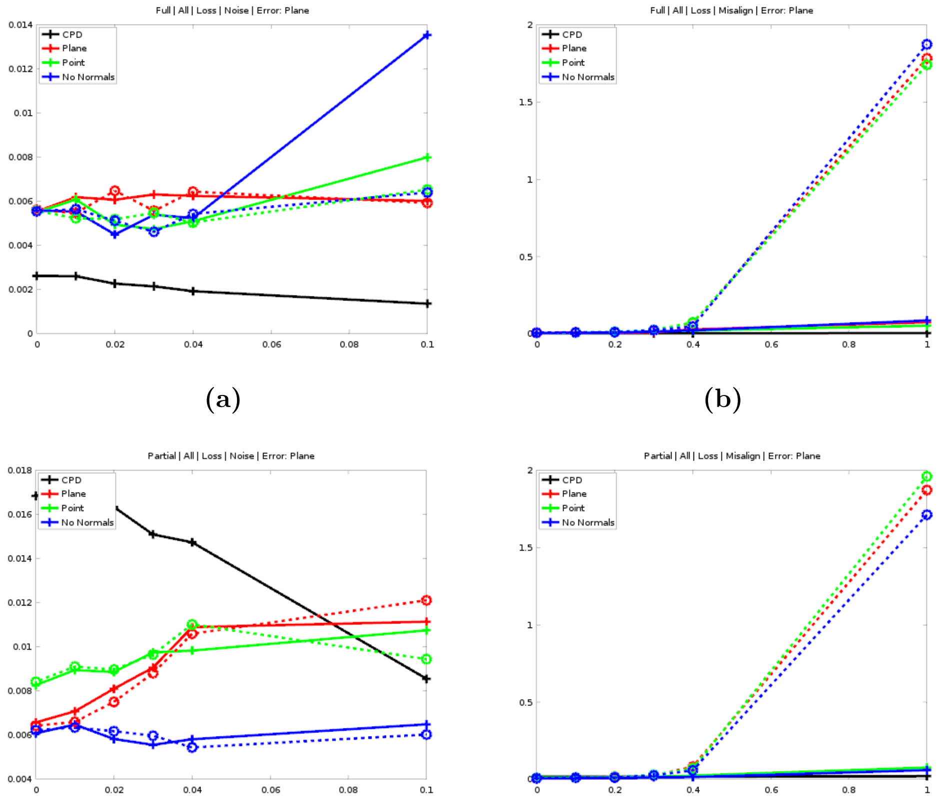
Partial | All | Loss | Noise | Error: Point

Partial | All | Loss | Misalign | Error: Point

(a) **(b)**

(c) **(d)**

Figure 4.4.: The same plots as in Fig. 4.2 but with the results of adding the Huber Loss function to each variant of our method. The loss function results are given by the dashed lines and circles, with the colors corresponding to the method used.



(a)

(b)

(c)

(d)

Figure 4.5.: The same plots as in Fig. 4.4 but with the average point-plane error instead.

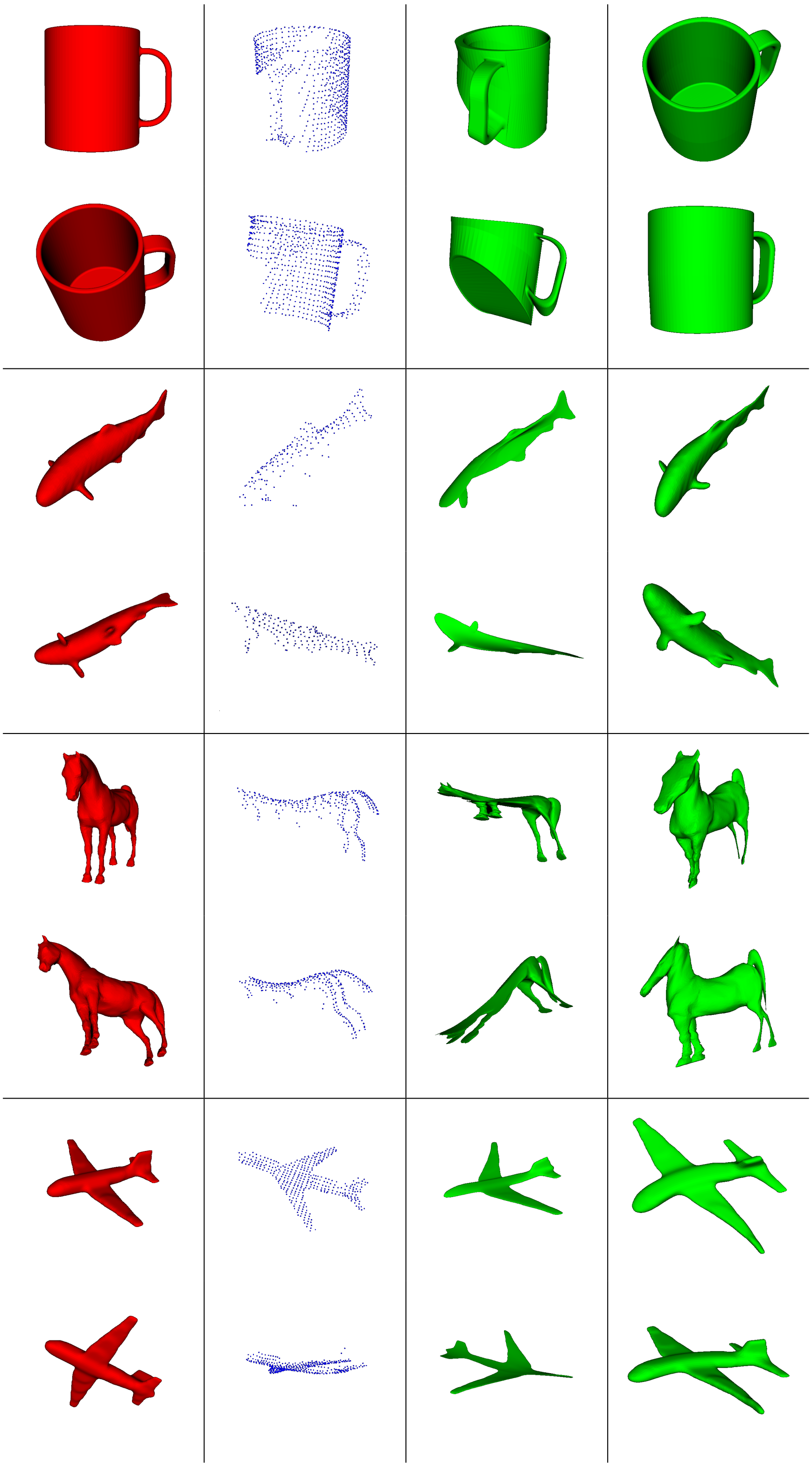


Figure 4.6.: A typical result from each of the classes when registering with partially-occluded shapes; each result is given from two different views. Left to right: canonical shapes, observed shapes, CPD's results, our method's results.

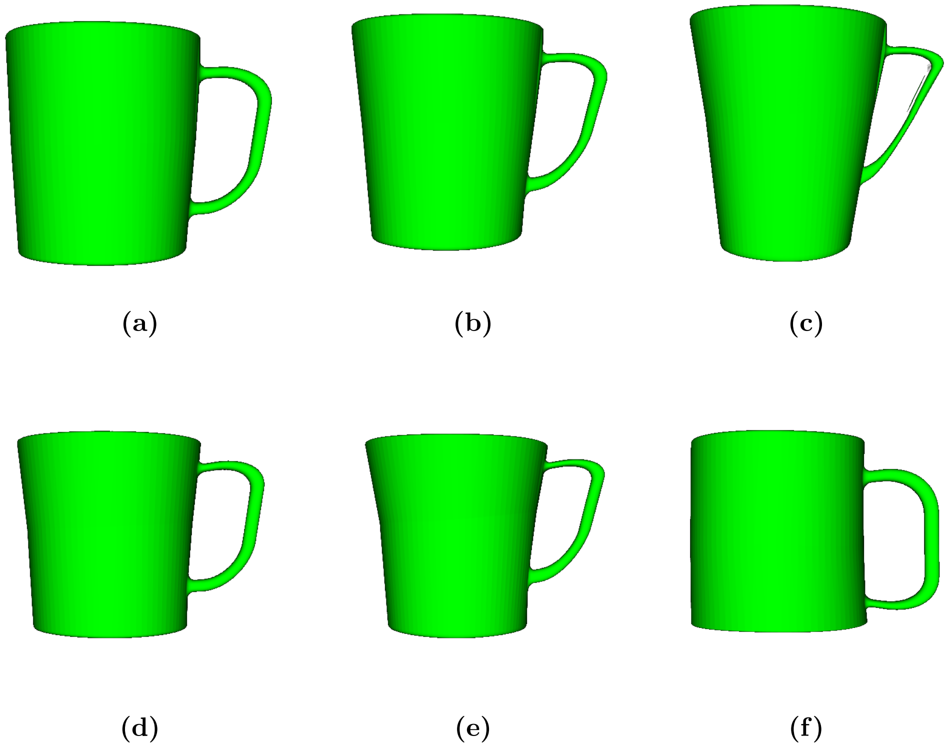


Figure 4.7.: Several novel shapes from the class *Mug* created from a transformation manifold with $q = 2$.

the other variants in most cases. Though the normal information does appear to assist in the case of extreme misalignment. The results of adding a robust loss function are given in the plots in Fig. 4.4 and Fig. 4.5. Again, the results were basically the same. However, it does add a degree of robustness to noise, but at a cost of a significant amount of robustness against extreme misalignment.

We also give a class-by-class error in Tab. 4.3 and Tab. 4.4. We noted that the method performed better when the range of deformations were low, as in the case of the class *Fish*, or when there were a large amount of training examples, as with the class *Mugs*. Because the method is limited to linear combinations of previously observed deformations, the method struggles when a class of objects has a lot of variety and not a lot of training examples, as was the case with the class *Animals*, or when the results of CPD were not optimal, as was the case with the class *Planes*. However, in the case partial views, our method consistently had less error than direct CPD.

The transformation manifold learned in the training phase can also be used to create transformations resulting in plausible novel shapes. We demonstrate a few

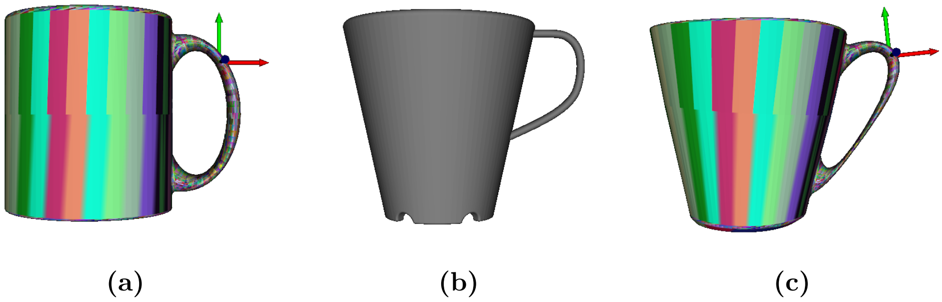


Figure 4.8.: Texture, segmentations, and grasp poses can be transferred from (a) to (b); the results of this transfer are seen in (c). Segmentation information (not visible) are attached to the vertices in the mesh.

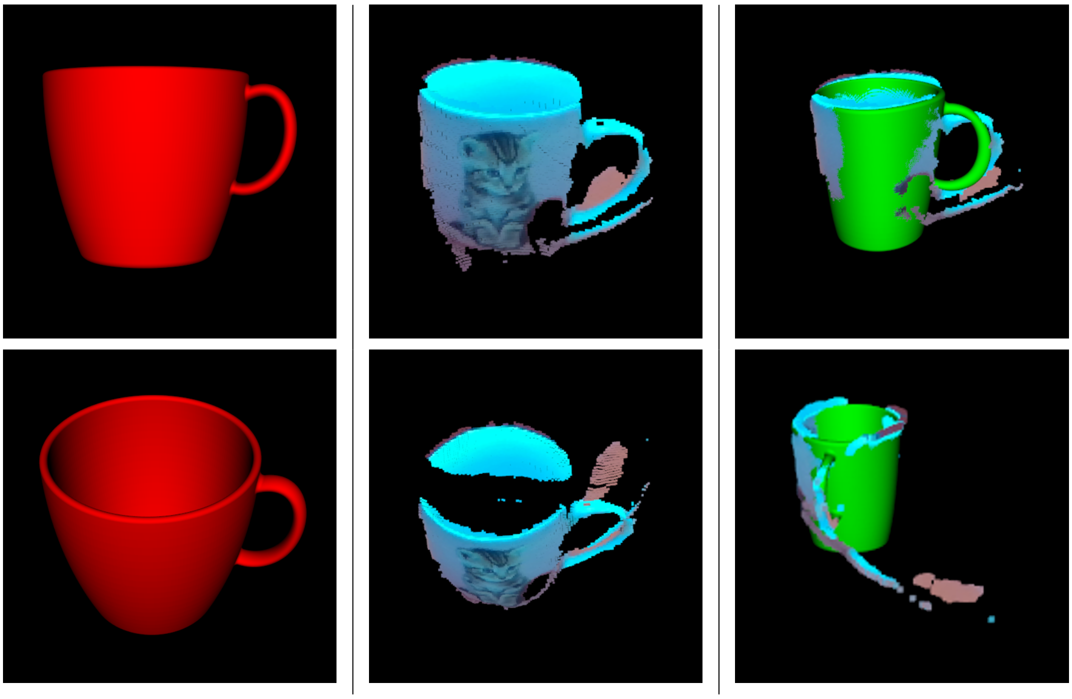


Figure 4.9.: Our method's results on a real mug with data acquired by a depth camera. Left to right: the canonical shape, the self-occluded and noisy observed shape, and the transformed canonical shape overlayed onto the observed shape.

of those shapes in Fig. 4.7. We also illustrate that by choosing different canonical shapes or annotating it, things such as high frequency features, segmentations, textures, and semantic information can also be transferred to a novel instance, such as in Fig. 4.8. And finally we show our method working on real world data captured with a depth camera in Fig. 4.9.

5. Conclusion

5.1. Conclusion

In this work, we presented a novel learned non-rigid registration method, which works for partially- or fully-observed shapes. We demonstrated its robustness to noise and occlusion, and gave several examples of applications such as manipulation skill transfer or generating novel instances.

We evaluated our method on several sets of shapes from the SCHREC 2007 dataset and an online CAD repository, GrabCab. We found that while the method performed slightly worse than conventional CPD on noiseless, fully observed shapes, when shapes were partially occluded, our method was much more able to recover the true underlying shape and reported lower error in both point-to-point and point-to-plane average distances.

We also presented several variants to our method and their results. With the current implementation, we feel the extra normal information does not appear to make enough of an impact to justify the extra space and computation time, and in many cases it performed worse than not using the information at all. The use of a loss function can slightly improve the robustness of the method against noise, but severely restricts its robustness against more extreme misalignment. A different robust loss function might overcome this limitation, such as one that adapts to the average point distance; we leave this as future work.

5.2. Future Work

Currently the method requires a coarse initial alignment. By adding a stage to compute coarse correspondences via features, we could incorporate a global alignment phase into the method. Additionally, the method is restricted by the diversity and size of the training data. We think a composite method combining CPD or another general non-rigid registration method and our method may produce better inference results. Also, while our initial attempts at incorporating surface information in the form of normals did not produce significantly better results than without, we would also like to continue pursuing enhancing the method to perform more surface-to-surface matching rather than the point-to-point matching it does now.

Appendix

A. Rodriguez Formula

A point v can be rotated by the equation:

$$v_{\text{rot}} = v\text{cos}\theta + (k \times v)\text{sin}\theta + k(k \cdot v)(1 - \text{cos}\theta) \tag{1}$$

where v_{rot} denotes the rotated point, k the axis of rotation, and θ the angle of rotation.

B. Estimation of Surface Normals

Algorithm 3: Estimation of a Surface Normal Given a Set of Neighbors $\mathbf{X}$

```
Input: $\mathbf{X} = (x_1, ..., x_n)^T$
    1. Compute plane centroid: $m = \frac{1}{n}\sum_{i=1}^{n} x_i$
    2. Subtract the centroid from the points: $x_i^* = x_i - m$
    3. Compute SVD of $\mathbf{X}^*$: $\mathbf{X}^* = \mathbf{USV}^T$
    4. A normal to the plane is given by the last column of U: $\hat{n} = U[:, last]$
    5. Flip the normal if necessary ($\hat{n} = -\hat{n}$)
Output: $\hat{n}$
```

C. Rotations

While the transformation model does not directly model point rotations, we can estimate the rotation of a point by formulating it as Wahba's Problem and solving it via singular value decomposition (SVD) ("Attitude determination using vector observations and the singular value decomposition" n.d.). We can estimate the rotation matrix $\mathbf{R}$ of the point by first creating a set of neighboring points $\mathbf{W} = (w_1, ..., w_k)^T$ around the point of interest, then apply the transformation model to the points to get their transformed points $\mathbf{V} = (v_1, ..., v_k)^T$.

We first obtain the matrix $\mathbf{B}$:

$$\mathbf{B} = \sum_{i=1}^{k} w_i v_i^T$$

Then we compute the singular value decomposition of $\mathbf{B}$:

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

And then the rotation matrix $\mathbf{R}$ is given by the equation:

$$\mathbf{R} = \mathbf{U}\mathbf{M}\mathbf{V}^T$$

where

$$\mathbf{M} = \mathrm{diag}([1 \quad 1 \quad \det(\mathbf{U})\det(\mathbf{V})])$$

Bibliography

Allen, B., B. Curless, and Z. Popović (2003). "The space of human body shapes: reconstruction and parameterization from range scans". In: *ACM transactions on graphics (TOG).* Vol. 22. 3. ACM, pp. 587–594 (cit. on pp. 7, 9).

Anguelov, D., P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis (2004). "The correlated correspondence algorithm for unsupervised registration of non-rigid surfaces." In: (cit. on p. 7).

"Attitude determination using vector observations and the singular value decomposition" (n.d.). In: (cit. on p. 37).

Bernard, F., F. R. Schmidt, J. Thunberg, and D. Cremers (2016). "A Combinatorial Solution to Non-Rigid 3D Shape-to-Image Matching". In: *arXiv preprint arXiv:1611.05241* (cit. on p. 8).

Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford university press (cit. on p. 14).

Blanz, V. and T. Vetter (1999). "A morphable model for the synthesis of 3D faces". In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques.* ACM Press/Addison-Wesley Publishing Co., pp. 187–194 (cit. on pp. 7, 9).

Bolkart, T. and S. Wuhrer (2015). "A groupwise multilinear correspondence optimization for 3d faces". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3604–3612 (cit. on p. 7).

— (2016). "A Robust Multilinear Model Learning Framework for 3D Faces". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4911–4919 (cit. on p. 7).

Bronstein, A. M., M. M. Bronstein, and R. Kimmel (2006). "Efficient computation of isometry-invariant distances between surfaces". In: *SIAM Journal on Scientific Computing* 28.5, pp. 1812–1836 (cit. on p. 7).

Brown, B. J. and S. Rusinkiewicz (2004). "Non-rigid range-scan alignment using thin-plate splines". In: *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on.* IEEE, pp. 759–765 (cit. on p. 7).

— (2007). "Global non-rigid alignment of 3-D scans". In: *ACM Transactions on Graphics (TOG).* Vol. 26. 3. ACM, p. 21 (cit. on pp. 7, 8).

Burghard, O., A. Berner, M. Wand, N. Mitra, H.-P. Seidel, and R. Klein (2013). "Compact part-based shape spaces for dense correspondences". In: *arXiv preprint arXiv:1311.7535* (cit. on pp. 9, 10).

Chang, W. and M. Zwicker (2009). “Range scan registration using reduced deformable models”. In: *Computer Graphics Forum*. Vol. 28. 2. Wiley Online Library, pp. 447–456 (cit. on pp. 7, 8).

Chen, Y. and G. Medioni (1992). “Object modelling by registration of multiple range images”. In: *Image and vision computing* 10.3, pp. 145–155 (cit. on pp. 7, 13).

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38 (cit. on p. 17).

Engelmann, F., J. Stückler, and B. Leibe (2016). “Joint Object Pose Estimation and Shape Reconstruction in Urban Street Scenes Using 3D Shape Priors”. In: *German Conference on Pattern Recognition*. Springer, pp. 219–230 (cit. on pp. 9, 10).

GrabCad (2017). https://grabcad.com/library (cit. on p. 23).

Haehnel, D., S. Thrun, and W. Burgard (n.d.). “An extension of the ICP algorithm for modeling nonrigid objects with mobile robots”. In: (cit. on p. 7).

Hasler, N., C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel (2009). “A statistical model of human pose and body shape”. In: *Computer Graphics Forum*. Vol. 28. 2. Wiley Online Library, pp. 337–346 (cit. on pp. 7, 9).

Huang, Q.-X., G.-X. Zhang, L. Gao, S.-M. Hu, A. Butscher, and L. Guibas (2012). “An optimization approach for extracting and encoding consistent maps in a shape collection”. In: *ACM Transactions on Graphics (TOG)* 31.6, p. 167 (cit. on p. 9).

Huber, P. J. et al. (1964). “Robust estimation of a location parameter”. In: *The Annals of Mathematical Statistics* 35.1, pp. 73–101 (cit. on p. 22).

Izadi, S., R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison, and A. Fitzgibbon (2011). “Kinectfusion: real-time dynamic 3d surface reconstruction and interaction”. In: *ACM SIGGRAPH 2011 Talks*. ACM, p. 23 (cit. on p. 8).

Jayanti, S., Y. Kalyanaraman, N. Iyer, and K. Ramani (2006). “Developing an engineering shape benchmark for CAD models”. In: *Computer-Aided Design* 38.9, pp. 939–953 (cit. on p. 23).

Kim, V. G., Y. Lipman, and T. Funkhouser (2011). “Blended intrinsic maps”. In: *ACM Transactions on Graphics (TOG)*. Vol. 30. 4. ACM, p. 79 (cit. on p. 7).

Lévy, B., S. Petitjean, N. Ray, and J. Maillot (2002). “Least squares conformal maps for automatic texture atlas generation”. In: *Acm transactions on graphics (tog)*. Vol. 21. 3. ACM, pp. 362–371 (cit. on p. 7).

Li, H., B. Adams, L. J. Guibas, and M. Pauly (2009). “Robust single-view geometry and motion reconstruction”. In: *ACM Transactions on Graphics (TOG)*. Vol. 28. 5. ACM, p. 175 (cit. on p. 8).

Li, H., R. W. Sumner, and M. Pauly (2008). “Global Correspondence Optimization for Non-Rigid Registration of Depth Scans”. In: *Computer graphics forum*. Vol. 27. 5. Wiley Online Library, pp. 1421–1430 (cit. on p. 8).

Li, H., E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev (2013). "3D self-portraits". In: *ACM Transactions on Graphics (TOG)* 32.6, p. 187 (cit. on p. 8).

McInerney, T. and D. Terzopoulos (1996). "Deformable models in medical image analysis: a survey". In: *Medical image analysis* 1.2, pp. 91–108 (cit. on p. 8).

Myronenko, A. and X. Song (2010). "Point set registration: Coherent point drift". In: *IEEE transactions on pattern analysis and machine intelligence* 32.12, pp. 2262–2275 (cit. on pp. 4, 7, 13).

Newcombe, R. A., D. Fox, and S. M. Seitz (2015). "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 343–352 (cit. on pp. 8, 10).

Nguyen, A., M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas (2011). "An optimization approach to improving collections of shape maps". In: *Computer Graphics Forum*. Vol. 30. 5. Wiley Online Library, pp. 1481–1491 (cit. on p. 9).

Oikonomidis, I., N. Kyriazis, and A. A. Argyros (n.d.). "Efficient model-based 3D tracking of hand articulations using Kinect." In: (cit. on p. 7).

Ovsjanikov, M., Q. Mérigot, F. Mémoli, and L. Guibas (2010). "One point isometric matching with the heat kernel". In: *Computer Graphics Forum*. Vol. 29. 5. Wiley Online Library, pp. 1555–1564 (cit. on p. 7).

Qian, C., X. Sun, Y. Wei, X. Tang, and J. Sun (2014). "Realtime and robust hand tracking from depth". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1106–1113 (cit. on p. 7).

Rohlfing, T. and C. R. Maurer (2003). "Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees". In: *IEEE Transactions on Information Technology in Biomedicine* 7.1, pp. 16–25 (cit. on p. 8).

Rohlfing, T., C. R. Maurer, W. G. O'dell, and J. Zhong (2004). "Modeling liver motion and deformation during the respiratory cycle using intensity-based nonrigid registration of gated MR images". In: *Medical physics* 31.3, pp. 427–432 (cit. on p. 8).

Roweis, S. (n.d.). "EM algorithms for PCA and SPCA". In: (cit. on p. 16).

Rueckert, D., A. F. Frangi, and J. A. Schnabel (2003). "Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration". In: *IEEE transactions on medical imaging* 22.8, pp. 1014–1025 (cit. on p. 8).

Rueckert, D., L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes (1999). "Nonrigid registration using free-form deformations: application to breast MR images". In: *IEEE transactions on medical imaging* 18.8, pp. 712–721 (cit. on p. 8).

Rusinkiewicz, S. and M. Levoy (2001). "Efficient variants of the ICP algorithm". In: *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, pp. 145–152 (cit. on p. 7).

Stückler, J. and S. Behnke (2014). “Multi-resolution surfel maps for efficient dense 3D modeling and tracking”. In: *Journal of Visual Communication and Image Representation* 25.1, pp. 137–147 (cit. on p. 9).

Stueckler, J., R. Steffens, D. Holz, and S. Behnke (2011). “Real-Time 3D Perception and Efficient Grasp Planning for Everyday Manipulation Tasks.” In: *ECMR*, pp. 177–182 (cit. on pp. 1, 9).

Süßmuth, J., M. Winter, and G. Greiner (2008). “Reconstructing Animated Meshes from Time-Varying Point Clouds”. In: *Computer Graphics Forum*. Vol. 27. 5. Wiley Online Library, pp. 1469–1476 (cit. on p. 8).

Tevs, A., M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel (2009). “Isometric registration of ambiguous and partial data”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, pp. 1185–1192 (cit. on p. 7).

Tipping, M. E. and C. M. Bishop (1999a). “Mixtures of probabilistic principal component analyzers”. In: *Neural computation* 11.2, pp. 443–482 (cit. on p. 17).

— (1999b). “Probabilistic principal component analysis”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3, pp. 611–622 (cit. on p. 17).

Wand, M., B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling (2009). “Efficient reconstruction of nonrigid shape and motion from real-time 3D scanner data”. In: *ACM Transactions on Graphics (TOG)* 28.2, p. 15 (cit. on p. 8).

Wand, M., P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling (2007). “Reconstruction of deforming geometry from time-varying point clouds”. In: (cit. on p. 8).

Yang, J., H. Li, D. Campbell, and Y. Jia (2016). “Go-ICP: a globally optimal solution to 3D ICP point-set registration”. In: *IEEE transactions on pattern analysis and machine intelligence* 38.11, pp. 2241–2254 (cit. on p. 7).

Yuille, A. L. and N. M. Grzywacz (1988). “The motion coherence theory”. In: *Computer Vision., Second International Conference on*. IEEE, pp. 344–353 (cit. on p. 13).

Zeng, Y., C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios (2010). “Dense non-rigid surface registration using high-order graph matching”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, pp. 382–389 (cit. on p. 7).

Zollhöfer, M., M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, et al. (2014). “Real-time non-rigid reconstruction using an RGB-D camera”. In: *ACM Transactions on Graphics (TOG)* 33.4, p. 156 (cit. on p. 8).