Modelling 3D Object Shape

Charlie Nash, Christopher K.I. Williams

The University of Edinburgh

Overview

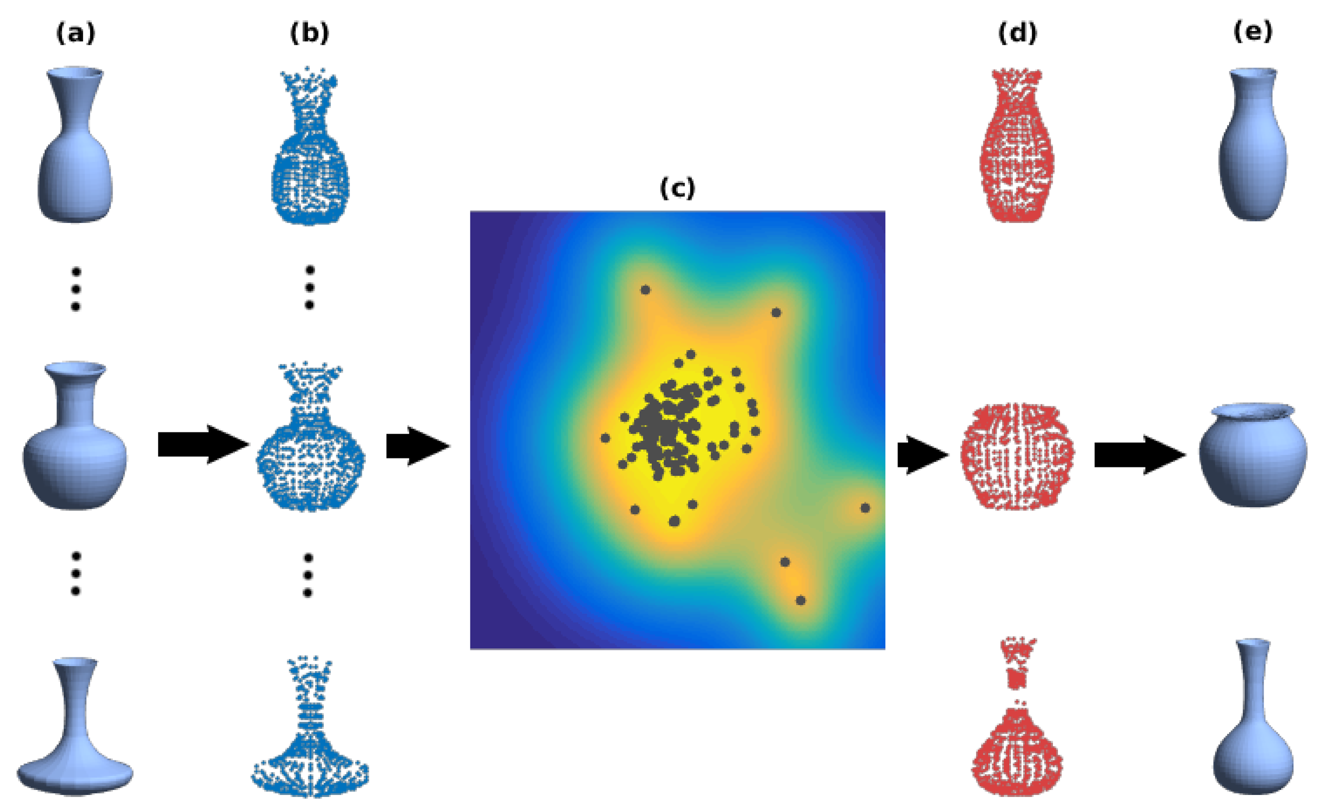


Figure 1: System overview. **(a)** We take a collection of 3D meshes from a particular object class. **(b)** Corresponding landmark points are automatically obtained. **(c)** We learn a latent space model using the landmark point representation. **(d)** We use the learned model to sample new instances from the object class in the form of landmark points. **(e)** We then generate a mesh that matches the shape of the sampled landmark points.

Motivation

- A model of object shape can be very useful for computer vision applications, whether as a means of generating richly-annotated training data for a recognition model, or as a component in an inverse-graphics system.
- Additionally content creation is a central task in computer graphics, and a shape model can be used to synthesize realistic objects that can be placed within a scene.

Aim

- In this project we aim to develop a system that can generate novel instances of an object class, using a collection of examples from that object class as training data.

Establishing Point Correspondences

Shape representation

- We use a landmark shape representation. Three-dimensional shapes can be represented by a collection of n points $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^{n}$, where $\mathbf{x}_i \in \mathbb{R}^3$, $i = 1, \ldots, n$.

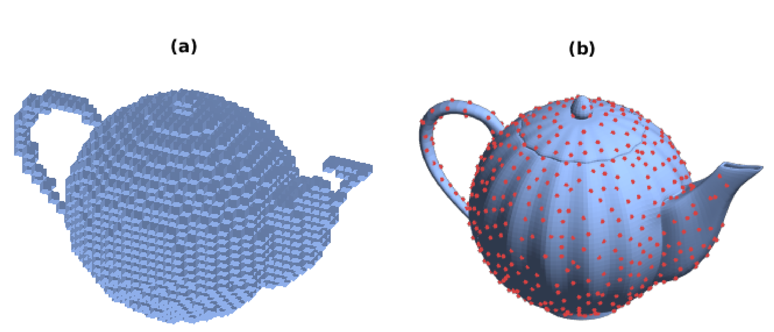


Figure 2: Representations of object shape. **(a)** Voxellated teapot model. **(b)** Teapot mesh with shape represented by dense landmark points.

Point correspondences

- Point correspondences (Figure 3) are a requirement for surface shape modelling.
- We deform a set of *template* landmarks so that they match a *target* mesh (Figure 4).

Figure 3: Correspondences obtained for two car models.

Optimisation

- We deform the template landmarks by minimising the following objective function:

$$E(\Psi) = E_{\mathrm{D}}(\Psi) + \alpha E_s(\Psi), \text{ where:}$$

$$E_{\mathrm{D}}(\Psi) = \sum_{i=1}^{N} w_i \left\| \mathrm{A}_i \mathbf{x}_i - \mathbf{v}_{n(i)} \right\|^2 + \lambda \sum_{j=1}^{M} w'_j \left\| \mathrm{A}_{m(j)} \mathbf{x}_{m(j)} - \mathbf{u}_j \right\|^2,$$

$$E_s(\Psi) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}(i)} \left\| (\mathrm{A}_i - \mathrm{A}_j) \mathrm{G} \right\|_F^2.$$

- The data fit term encourages transformed template points to be close to target vertices.
- The regularisation term encourages transformations of neighbouring landmarks to be similar.
- The energy is minimised using the iterative closest point algorithm, which alternately finds correspondences between template and target points, and then minimizes for the current correspondences.
- The stiffness constant α is reduced gradually over the course of the optimisation.

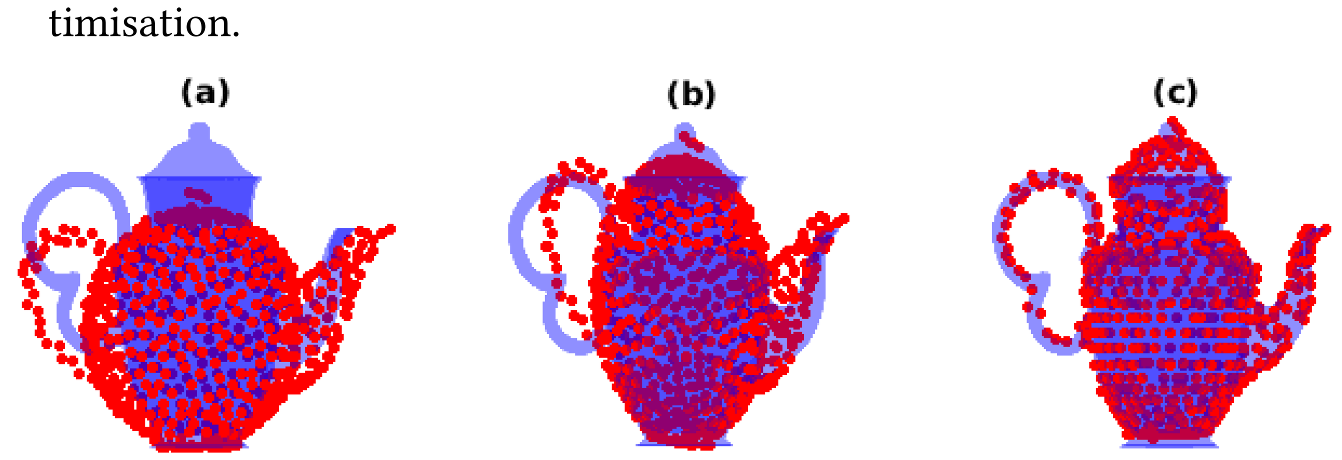


Figure 4: Template deformation. **(a)** Target model in blue with template landmark points shown in red. **(b)** Intermediate deformation. **(c)** Template fully deformed into the target mesh.

Models for object shape

Latent variable models

- In order to model object shape we specify a generative model $p(\mathbf{x}|\boldsymbol{\theta})$ and learn the parameters $\boldsymbol{\theta}$ from a collection of corresponding landmark points $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{M}$ for a particular object class.
- We make use of the low inherent-dimensionality of the data with latent variable models: PPCA, GP-LVM and Bayesian GP-LVM models.

Shape manifold

- Figure 5 shows how tracing a straight line in latent space results in a smooth morph in data space.

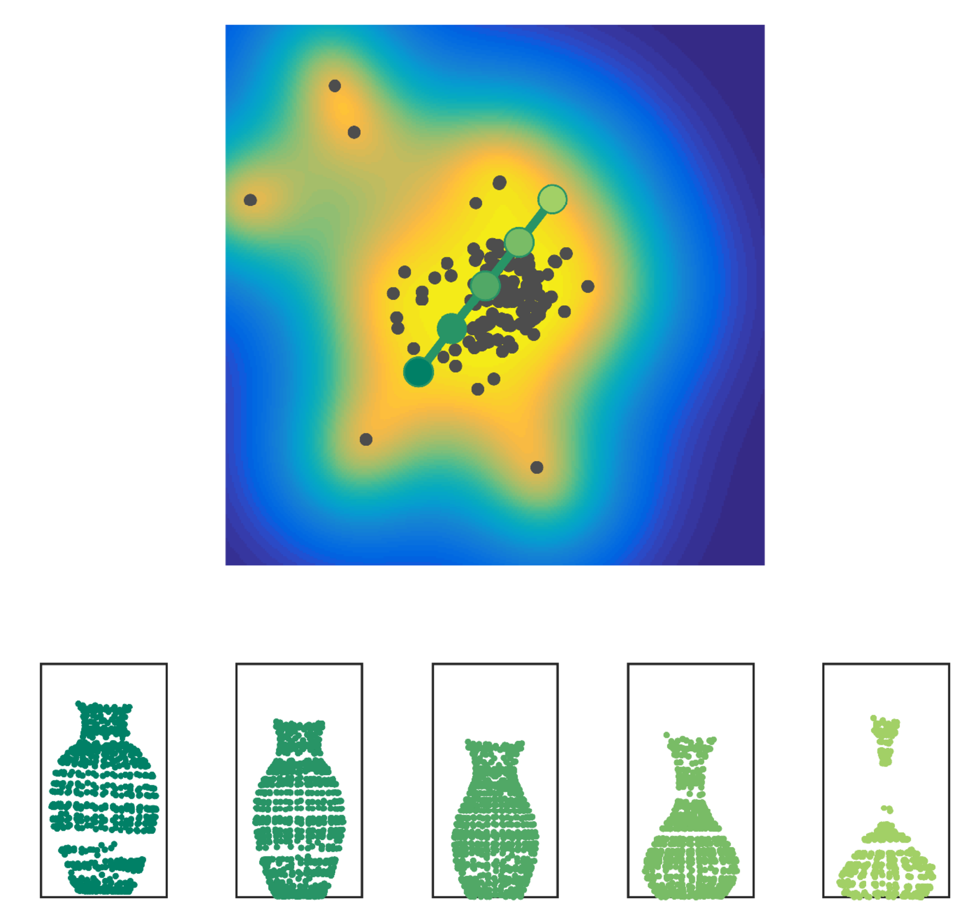


Figure 5: Smooth data morphing using the GP-LVM. The colormap encodes the precision with which the manifold is represented in data space at each point in latent space.

Shape completion

- The generalisation capability of the shape models can be tested on a shape completion task.
- An unseen example is chosen and a number of its variables are masked. The task is then to predict these variables given the remaining visible variables.

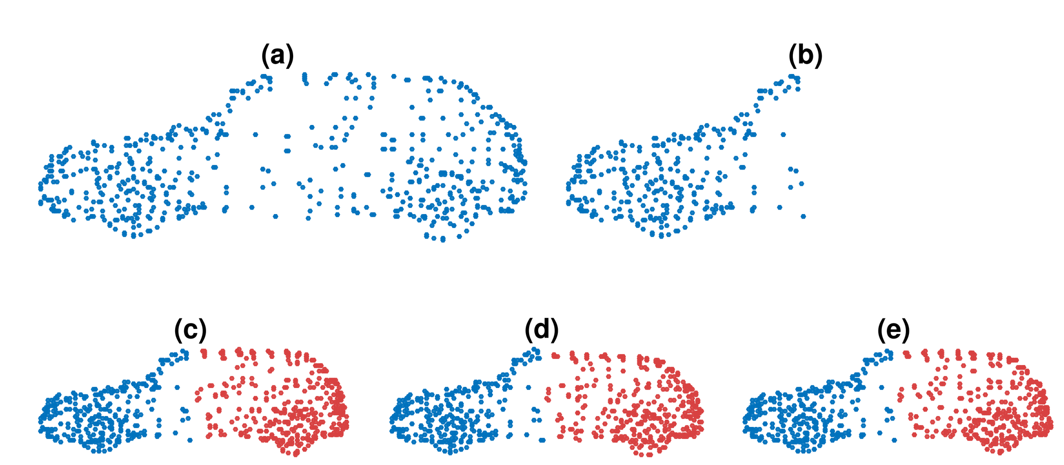


Figure 6: Shape completion. **(a)** Test shape from cars dataset. **(b)** Test shape with missing points. **(c)** PPCA reconstruction. **(d)** GP-LVM reconstruction. **(e)** Bayesian GP-LVM reconstruction.

Meshing shape samples

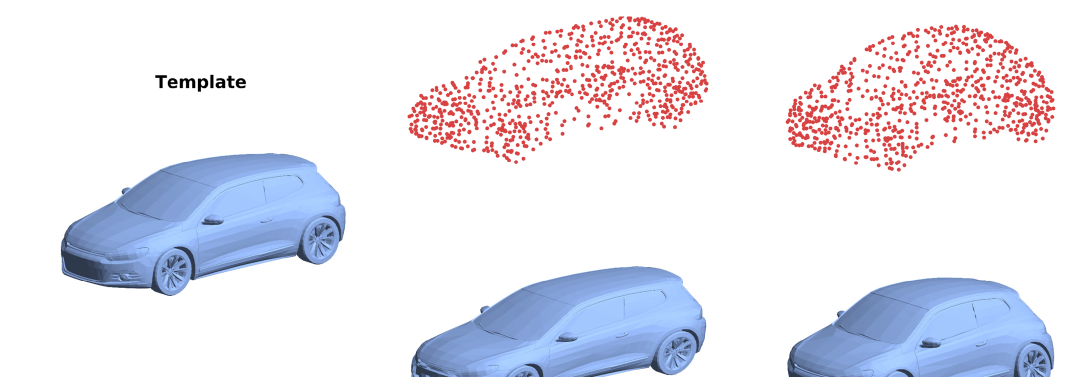


Figure 7: Meshing samples. The landmarks are generated by the shape model and the embedded deformation is applied to a template shape to generate the meshes.

Template transformation

- Given a new shape sample we deform a template model so that its shape matches the sample (Figure 7).
- We find affine transformations $\{\mathrm{A}_i\}_{i=1}^{N}$ that map the template landmarks $\{\mathbf{x}_i\}_{i=1}^{N}$ to the sample landmarks $\{\mathbf{u}_i\}_{i=1}^{N}$ by minimizing an energy function with data fit term:

$$E_{\mathrm{D}}(\Psi) = \sum_{i=1}^{N} w_i \left\| \mathrm{A}_i \mathbf{x}_i - \mathbf{u}_i \right\|^2.$$

Embedded deformation

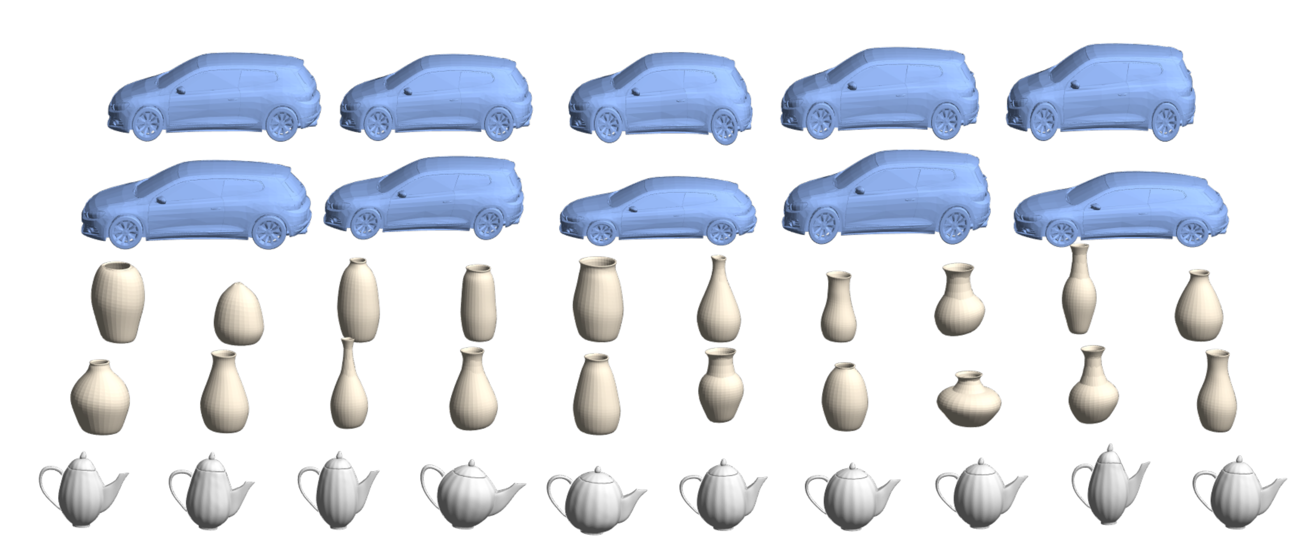
- The template vertices are then deformed using an embedded deformation:

$$\tilde{\mathbf{v}}_j = \sum_{i=1}^{N} w_j(\mathbf{v}_j)[\mathrm{A}_i(\mathbf{v}_j - \mathbf{x}_i) + \mathbf{x}_i],$$

$$w_i(\mathbf{v}_j) = \begin{cases} (1 - \left\| \mathbf{v}_j - \mathbf{g}_i \right\| / d_{\max})^2 & \text{if } \mathbf{g}_i \in \mathrm{kNN}(\mathbf{v}_i) \\ 0 & \text{otherwise.} \end{cases}$$

- Given a synthesised mesh, we optionally apply curvature flow smoothing to remove rough features and noise.
- When determining the affine transformations $\{\mathrm{A}_i\}_{i=1}^{N}$ the stiffness term can be varied so as to balance faithfulness to the sampled land marks and mesh quality.

Summary



- Our method automatically finds corresponding landmark points on shapes within an object class. A latent-variable model is used to generate new sets of landmark points. A template mesh is deformed which matches the shape of the sampled landmark points.
- Our method generates novel and realistic samples from an object class with minimal manual intervention.