Single and Sparse View 3D Reconstruction by Learning Shape Priors

Yu Chen and Roberto Cipolla

Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK

Abstract

In this paper, we aim to reconstruct free-form 3D models from only one or few silhouettes by learning the prior knowledge of a specific class of objects. Instead of heuristically proposing specific regularities and defining parametric models as previous research, our shape prior is learned directly from existing 3D models under a framework based on the Gaussian Process Latent Variable Model (GPLVM). The major contributions of the paper include: 1) a framework for learning the shape prior of the 3D objects, which requires no heuristic of the object, and can be easily generalized to handle various categories of 3D objects, and 2) novel probabilistic inference schemes for automatically reconstructing 3D shapes from the silhouette(s) in the single view or sparse views. Qualitative and quantitative experimental results on both synthetic and real data demonstrate the efficacy of our new approach.

Keywords: single view reconstruction, shape-from-silhouettes, shape priors

1. Introduction

Reconstructing 3D shapes from 2D images can be a hard problem if limited inputs are provided. Typical examples include single view reconstruction (SVR) and shape-from-silhouettes (SFS) in the sparse-view setting (see Fig. 1). In these cases, the reconstruction problem becomes severely under-constrained. Few or no reliable image correspondence is available for setting up the stereo framework, while available geometrical clues, such as silhouettes, depth maps, and normal maps, are usually far from enough to obtain an unambiguous reconstruction of the model.

In view of this problem, previous research makes strong assumptions and proposes shape priors of either general or specific 3D objects/scenes, e.g., minimizing overall smoothness [1], planar or ground-vertical scenes [2], to further constrain the problem. A common major drawback of these methods is that those priors are mainly defined by heuristics and suitable only for special cases. Such limitation prevents these methods from reconstructing those models with more complex and curved geometry like human faces or human bodies.

Another line of research is to manually define parametric models for each specific category, such as 3D faces [3] and human bodies [4], [5], in order to characterize the variation or the morphing of the 3D shape. The model parameters can thus be learned from the training data. The main drawback of these approaches

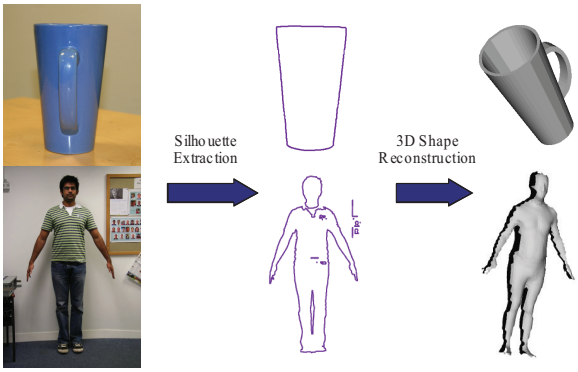


Figure 1: Reconstructing the 3D shapes of a mug and a human body from single 2D image/silhouette inputs. The results are generated by the approach proposed in the paper.

is that parametric models are only suitable for describing the shape of limited categories of objects, and it is hard to generalize the model for reconstructing objects in other categories.

In this paper, we make an attempt to learn the shape knowledge from existing 3D models instead of proposing specific reconstruction rules and shape models from heuristics. We observe that the shapes of the same class of objects are actually controlled by a small number of factors notwithstanding the complex geometrical or topological structures. By manipulating them, we can then model the 3D shape of the whole class of ob-

jects with much fewer parameters and the detailed reconstruction can be estimated from the 2D image with much less difficulty.

Hence, our goal is to extract these factors automatically in the learning process, which is more generalized compared with the parametric shape models in the past literatures, no prior knowledge from the shape is assumed in the proposed framework, and the latent factors that control the shape variation of the objects are also treated unknown in advance. In this paper, Gaussian Process Latent Variable Model (GPLVM) [6] is used to complete this task, i.e., to extract the unknown low dimensional embedded information of the object from high dimensional observations given a relatively small amount of training samples. The GPLVM and its variants have been applied to solve computer vision problems, mainly in the context of human pose estimation [7], [8], [9], [10] and tracking the deformable surface [11]. Compared with widely shape modeling approaches such as PCA [4], [5], GPLVM not only gives a much more compact representation, but also captures the multi-modality of the solution and provides an uncertainty measurement of the prediction.

Our framework requires little interaction and it can be generalized to reconstruct various categories of 3D objects which may have more complex structures. Experiments performed on the synthetic and real examples show that our new approach is plausible even though only one or few 2D silhouettes are given as inputs.

The rest of this paper is organized as follows. A brief review on previous methods on single view reconstruction is given in Section 2; the framework of our learning-based reconstruction and detailed techniques involved are presented in Section 3; experimental results are provided in Section 4; related discussions are given in Section 5; and finally, a brief conclusion is drawn in Section 6.

2. Related Work

Single view reconstruction and shape-from-silhouettes are popular topics in the area of computer vision and much investigated in the previous literatures.

Pure geometrical methods are the main streams of current research on SVR. Much research has been done on planar outdoor architecture scenes. Criminisi et al. [12] recover the 3D affine geometry from a single perspective image based on vanishing points information and projective geometry constraints. All the measurements in the scene can be accurately obtained once the scale factor is determined. Similar constraints are used by [13], [2]. Hoiem et al. [2] first segment the image into 3 categories: ground, sky, verticals. A coarse pop-up reconstruction is then obtained based on the segmentation results. Barinova et al. [13] further use a Conditional Random Field (CRF) model to infer the ground-vertical boundary parameters. Similar projective geometry-based approach is also generalized by Delage et al. [14] to reconstruct indoor scenes. All the studies above, however, only focus on planar scenes and also assume a ground verticality for the scene reconstruction. In contrast, Saxena et al. [15] investigate the relation between scene depths and image features using a Markov Random Field (MRF) model. The method, however, only gives a rough depth estimate of the scene.

Recent work in computer vision and computer-aided design has looked at the 3D shape reconstruction from 2D sketches or line drawings in the single view [16], [17], [18]. The objects are represented by edge-vertex graphs. The common methodology of the research is to propose regularities manually based on the geometrical or topological constraints of the line drawings that can represent a realistic object, and then formulate the reconstruction problem into an optimization problem with respect to the 3D positions of vertices or edges. Most of the research is limited to dealing with abstract objects, which are transparent (with hidden lines visible) and planar.

Some efforts have also addressed the problem of reconstructing curved or free-form objects [1], [19]. Zhang et al. [19] introduce several user constraints such as normal map, depth discontinuities, creases, etc, and finally formulate the reconstruction problem into linearly constrained quadratic optimization problem which has a closed-form solution. However, their method is only limited to generate 2.5D Monge patches. Prasad et al. [1] generalize the scope of the problem to full 3D surfaces. They basically adopt the framework of constraints-based optimization in [19]. Contour Generators (CGs) are used for creating curved patches and objects with more complicated topology are studied. However, their approach still requires subtle interaction to mark up the parametric space when the topology of the 3D object becomes complex. Due to the ambiguous nature of SVR problems, all these approaches above often require considerable amount of interaction from users and they are usually based on some heuristic regularities such as minimizing the overall smoothness.

Shape priors like symmetry, geometrical constraints or other scene cues are usually insufficient to model more elegant structure and subtle shape variation. In view of the problem, stronger priors of objects are investigated, usually through defining the parametric models

of a specific category, e.g., human bodies. The problem of reconstructing 3D shapes is transformed into learning the parameters of the shape model. A representative parametric model is SCAPE (Shape Completion and Animation for PEople) [4], which is a data-driven method for building body shapes with different articulation poses and individual shape variation. This model has recently been applied to estimating the human body shape from a monocular image [5], [20]. Sigal et al. [5] have proposed a discriminative model based on Mixture of Experts to recover the SCAPE model parameters as well as a generative stochastic optimization which helps refining the estimation. Their method allows the discriminative estimation of articulation pose and body shape directly from monocular and multi-camera image silhouettes. Guan et al. [20] extend their work on the SVR problem and improve the reconstruction by including the shading cues. The drawbacks of both methods are that they depend heavily on the accurate pose initialization and a clean segmentation, and also, the optimization of the model is based on sampling and searching in the parameter space, which can be extremely expensive. Compared with this work, our approach differs in that the proposed framework is targeted to model different categories of objects and more general shapes rather than designed for the specific category.

The problem of inferring the 3D shape of objects within a learning framework is beginning to receive attention. Representative research includes [21], [22], [23], [24]. These approaches are quite relevant to ours. In [24], Torresani et al. try to learn the time-varying shape of non-rigid 3D object from uncalibrated 2D tracking data, usually monocular video sequences which record the motion of a specific object. The shape distribution is assumed and the motion and deformation of the model are estimated through a generalized EM algorithm. The approach gives satisfying results on synthetic data and are robust to missing data. However, these algorithms are very slow even for relatively simple models. The difference between the goal of their approach and ours is that we focus more on learning the shape of a class of objects instead of tracking the motion of a specific object.

Han and Zhu [22] address a two-step Bayesian framework for representing the individual 3D shapes and modeling the global structure of the scene. Probabilistic inference is done by the Markov Chain Monte Carlo (MCMC) method with a number of reversible jumping rules defined. They have handled both man-made block objects and natural objects. However, there are two drawbacks of this framework. First, MCMC computation can be extremely time consuming; second, probabilistic shape priors of different categories of objects (such as prior models for polyhedra, grass, and trees in the paper) need to be manually defined before the model inference can be conducted.

Rother and Saprio [23] formulate the single view reconstruction problem as a belief inference problem on a hierarchical graphical model, which theoretically handles the pose estimation, recognition and reconstruction in a unified framework. However, how to obtain the shape priors of the 3D shapes remain unexplored in the paper.

Another related work is by Hassner and Basri [21], where the depth is reconstructed from examples, i.e., 3D geometries which look similar to the query object from a database serving as the shape prior. Their method is non-parametric and the depth of the query is synthesized with a hard-EM optimization on a target function based on patch similarities between every database instance and the query. The method has been proposed to reconstruct a wide variety of categorized objects, but it requires cleanly-segmented inputs and suffers from slow computation.

Our approach is similarly based on a learning framework. Different from the previous ones, we aim to learn the prior knowledge and rules of reconstruction from existing 3D models using statistical methods instead of proposing them empirically.

3. Approaches

Fig. 2 illustrates the framework of our approach, and it includes both procedures for training the shape prior model and predicting the 3D shape based on the model obtained. 2D silhouettes and the corresponding depth scans of 3D objects are used as training data. In the training stage (Fig. 2(a)), we first use a common shape template to encode the 2D position and depth information for each instance in the database (Section 3.1). After this, Principal Component Analysis (PCA) (Section 3.2) is applied to decorrelate and reduce the dimension of input data before training the shape prior using the GPLVM (Section 3.3).

In the prediction (reconstruction) stage, only 2D silhouettes are used. We propose two schemes for the shape recovery. The first scheme is a straightforward approach (Fig. 2(b)). The same preprocessing steps of registration and dimension reduction as the training stage are performed (Section 3.4). Then, depths of the object and uncertainty measurements are inferred by the GPLVM, which is trained from the combinational inputs of both 2D position and depth information. The

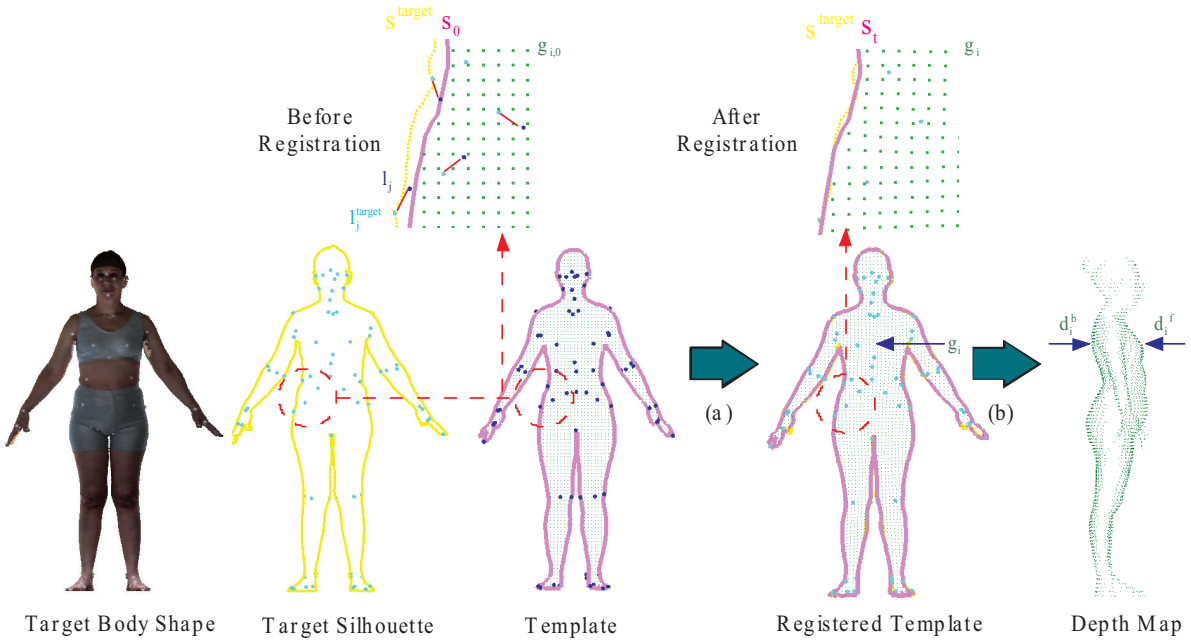


Figure 3: An illustration of the shape registration. (a) Template matching. Target and template silhouettes are represented by yellow and magenta curves, respectively; landmarks in the target and templated are marked by cyan and blue dots, respectively; template grid points are marked by green squares. (b) Depth maps extraction. Each 2D grid point $\mathbf{g}_i$ is associated with two depth samples d_i^f (frontal) and d_i^b (dorsal).

second scheme is a hierarchical model based on matching silhouettes with the shapes generated from the prior (Section 3.5). It allows the shape reconstruction from one or more noisy silhouette inputs and the correction of camera viewpoints.

3.1. Preprocessing and Registration

Registration of the 2D input and vectorizing the 2D position and depth information of the objects are necessary steps before the model training. For this purpose, a template matching scheme is applied in our approach. For each category of objects, the shape template with a deformable silhouette and a uniform grid inside is generated, as shown in Fig. 3. The motivation of using a template is that we find that the position information encoded by the grid is effective and less susceptible to the imperfectness and local distortion of input silhouettes.

Simply, we generate the template from an arbitrary instance in the category. In both training and testing stages, the template is deformed to fit the 2D shape of each instance by matching the silhouette and warping the internal grid points accordingly. A method based on the modified Iterative Closest Points (ICP) is adopted to efficiently match silhouettes in our approach. Then, the 2D position and depth information of that object are encoded by the displacements of these grid points. And finally, frontal and dorsal depth values are extracted at each grid point.

3.1.1. Silhouette Matching

We denote the deformable silhouette template in the t-th iteration as the point set $\mathbf{s_t} = \{\mathbf{s_{i,t}}\}_{i=1}^{N_s}$ and the target silhouette as the point set $\mathbf{s^{target}} = \{\mathbf{s_j^{target}}\}_{j=1}^{N_t}$ (see Fig. 3). The update equation at the t-th iteration can be written as:

$$\mathbf{s_{i,t+1}} = \mathbf{s_{i,t}} + \eta(\lambda_1 \mathbf{\Delta s^b_{i,t}} + \lambda_2 \mathbf{\Delta s^d_{i,t}} + \lambda_3 \mathbf{\Delta s^l_{i,t}}), \quad (1)$$

where the position update consists of the following three terms.

The first term $\mathbf{\Delta s^b_{i,t}}$, which is simply defined by the point-wise distance of two silhouettes, enforces the good matching between the template and the target silhouette, as shown in (2).

$$\mathbf{\Delta s^b_{i,t}} = \frac{\sum_{j=1}^{N_t} w_j^b (\mathbf{s_{i,t}} - \mathbf{s_j^{target}})}{\sum_{j=1}^{N_t} w_j^b}, \quad (2)$$

where $w_j^b = \exp(-\|\mathbf{s_{i,t}} - \mathbf{s_j^{target}}\|^2/\sigma^2)$, $j = 1, 2, \cdots, M$.

The second term $\mathbf{\Delta s^d_{i,t}}$ regulates that the neighboring points on the silhouette should maintain their relative

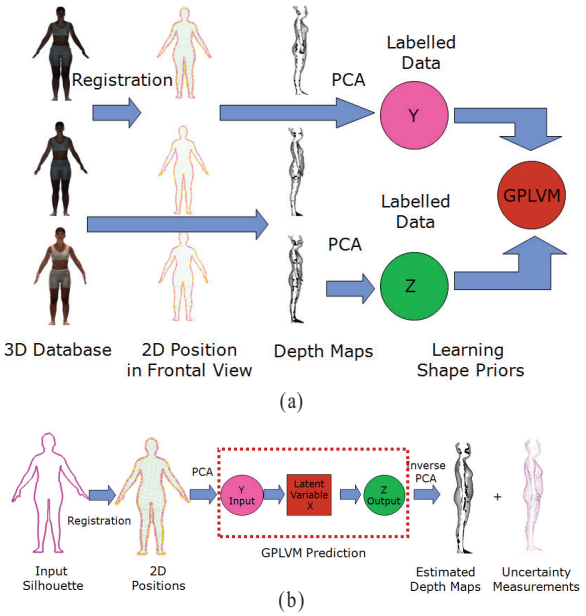


Figure 2: An overview of our approach: (a) training the shape prior; (b) reconstruction from single silhouette input.

positions from each other during the deformation. Ideally, they should neither cluster together nor stay far away from each other after the template deformation. It is given as:

$$\mathbf{\Delta s^d_{i,t}} = \frac{\sum_{j=1}^{N_s} w_j^d(\mathbf{s_{i,t}} - \mathbf{s_{j,t}} - \mathbf{s_{i,0}} + \mathbf{s_{j,0}})}{\sum_{j=1}^{N_s} w_j^d}, \tag{3}$$

where $w_j^d = \exp(-\|\mathbf{s_{i,0}} - \mathbf{s_{j,0}}\|^2/\sigma^2)$, $j = 1, 2, \cdots, N$.

In many 3D shape databases, landmarks are manually placed on the surface of the object to indicate critical positions of anthropometric or geometrical characteristics of the shape category. These landmarks are frequently used for setting up the correspondences and tracking the shape variation. The third term $\mathbf{\Delta s^l_{i,t}}$ ensures that the silhouette deformation should be coincident with the landmark registration of the template.

$$\mathbf{\Delta s^l_{i,t}} = \frac{\sum_{j=1}^{L} w_j^l(\mathbf{s_{i,t}} - \mathbf{l_j^{target}} - \mathbf{s_{i,0}} + \mathbf{l_j})}{\sum_{j=1}^{L} w_j^l}, \tag{4}$$

where $\{\mathbf{l_j}\}_{j=1}^L$ are the default positions of the landmarks in the template silhouette, $\{\mathbf{l_j^{target}}\}_{j=1}^L$ are the corresponding positions of those landmarks in the target silhouette, and the weighting factors $\gamma_j = \exp(-\|\mathbf{s_{i,0}} - \mathbf{l_j}\|^2/\sigma^2)$, $j = 1, 2, \cdots, L$. The landmark term is optional and valid only when landmark information is provided. However, we find it can be quite useful for fast initializing the positions of template silhouette points by setting $\lambda_1 = \lambda_2 = 0$ and $\lambda_3 = 1$ in the first iteration, and later guiding a quick and accurate silhouette matching.

The silhouette matching algorithm is run for several iterations with $\lambda_1 = \lambda_2 = \lambda_3 = 1$, $\eta = 0.5$, $\sigma = 0.02/4^t$ [1]. The parameter settings are fixed throughout all experiments.

3.1.2. Grid and Depth Map Generation

With the deformation of the silhouette, we also hope to establish a one-to-one mapping between the deformed template and the original one for all the grid points lying inside the silhouette.

We assume the grid density is N_g and consider the i-th sample grid point of the template. Let $\mathbf{g_i}$ and $\mathbf{g_{i,0}}$ be its after-warping position and its initial position before warping, respectively. To deform the grid, we first roughly estimate $\mathbf{g_i}$ with the silhouette displacement $\Delta\mathbf{g_s}$ and the landmark displacement (if landmark is given) $\Delta\mathbf{g_l}$, as shown in (5):

$$\begin{aligned}\mathbf{g_i} &= \mathbf{g_{i,0}} + \lambda'\Delta\mathbf{g_s} + (1-\lambda')\Delta\mathbf{g_l} \\ &= \mathbf{g_{i,0}} + \frac{\sum_{i=1}^{N_g} w_i^{s\prime}(\mathbf{s_{i,t}} - \mathbf{s_{i,0}})}{\sum_{i=1}^{N_g} w_i^{s\prime}} + \frac{\sum_{j=1}^{L} w_j^{l\prime}(\mathbf{l_j^{target}} - \mathbf{l_j})}{\sum_{j=1}^{L} w_j^{l\prime}},\end{aligned} \tag{5}$$

where $\{\mathbf{l}_j\}_{j=1}^L$ are the original positions of the landmarks in the template silhouette; $\{\mathbf{l_j^{target}}\}_{j=1}^L$ are the corresponding positions of those landmarks in the target silhouette; and weighting factors are $w_i^{s\prime} = \frac{\sigma'}{\|\mathbf{g_0} - \mathbf{s_{i,0}}\|}\exp(-\|\mathbf{g_0} - \mathbf{s_{i,0}}\|^2/\sigma'^2)$, $i = 1, 2, \cdots, N_g$ and $w_j^{l\prime} = \frac{\sigma'}{\|\mathbf{g_0} - \mathbf{l_j}\|}\exp(-\|\mathbf{g_0} - \mathbf{l_j}\|^2/\sigma'^2)$, $j = 1, 2, \cdots, L$, respectively. Here, we fix the parameters to be $\lambda' = 0.2$ and $\sigma' = 0.1$. Then, in order to generate a smoother and more homogeneous warping, we iteratively average the position of each grid point by its 4-neighborhood while pinpointing those grid points on the boundary.

Finally, the depth values are sampled from the 3D shape at the grid points of the registered template. Since the depth sampling is done on both sides of the object, two depth values d_i^f and d_i^b are extracted for the front and back for each 2D grid point $\mathbf{g_i}$. A grid with $2N_g$ 3D sample points is generated to represent the 3D shape. The complete procedure of the silhouette registration is given in Algorithm 1. We find the described registration method is efficient for clean but dense silhouettes and

[1] $\lambda_3 = 0$ is no landmark is provided.

Algorithm 1 Template-based silhouette registration algorithm.

1. Initialize the parameters $\lambda_1 = \lambda_2 = \lambda_3 = 1$, $\eta = 0.5$, $\sigma = 0.02$, $\lambda' = 0.2$, and $\sigma' = 0.1$.
2. **for** iteration $t = 1$ to T_1:
   (a) **If** $t = 1$, set $\lambda_1 = \lambda_2 = 0$; **else**, set $\lambda_1 = \lambda_2 = 1$.
   (b) Calculate $\mathbf{\Delta s^b_{i,t}}$, $\mathbf{\Delta s^d_{i,t}}$, and $\mathbf{\Delta s^l_{i,t}}$ (if any) for each sample point i.
   (c) Update the positions of template silhouette points $\mathbf{s_{i,t}}$, $i = 1, 2 \cdots, N_s$ with equation (1).
   (d) Update the parameter $\sigma = 0.02/2^t$.
3. Compute the internal warping and deform the grid with equation (5).
4. Smooth the grid by iteratively averaging the grid points in the 4-neighborhood.
5. Depth sampling at deformed grid points from both sides of the object.

grids. Empirically, the running time grows linearly with the resolution of the template grid.

3.2. Dimension Reduction and Decorralation

The raw position and depth data obtained from the template are not suitable for learning the GPLVM. First, each dimension of the original data, which corresponds to the coordinate of grid point, is highly correlated with the others, i.e., the coordinates of neighboring grid points. This fact violates the dimension independence assumption when training a GPLVM (see Section 3.3). Second, the high dimensionality (around 5000) of the original data requires huge memory consumption during the model training. Because of these problems we first decorrelate and compress the data using Principal Component Analysis (PCA).

For each input instance, we obtain the $2N_g$-D grid position vector $\mathbf{G_y} = [\mathbf{g_i}]_{i=1}^{N_g}$ and its corresponding $2N_g$-D depth-map vector $\mathbf{G_z} = [d_i^f, d_i^b]_{i=1}^{N_g}$ through the template matching. [2] They can be approximately represented into the linear combinations of mean vectors $\mathbf{G_{y,0}}$ and $\mathbf{G_{z,0}}$ of the training set, and the first m eigenvectors $\mathbf{A} = [\mathbf{a_i}]_{i=1}^m$ and $\mathbf{B} = [\mathbf{b_i}]_{i=1}^m$ of training-set covariance matrices, respectively, as the following equations show.

$$\mathbf{G_y} = \mathbf{G_{y,0}} + \sum_{i=1}^{m} y_i \mathbf{a_i} = \mathbf{G_{y,0}} + \mathbf{Ay} \tag{6}$$

$$\mathbf{G_z} = \mathbf{G_{z,0}} + \sum_{i=1}^{m} z_i \mathbf{b_i} = \mathbf{G_{z,0}} + \mathbf{Bz} \tag{7}$$

where the linear coefficients y_i and z_i can be used to characterize the 3D shape of the new instance. For our experiment, we use the first $m = 30$ principal components of both 2D positions and the depth maps, respectively, and they account for over 98% variance of datasets we investigate.[3] For each instance, m-D PCA feature vectors $\mathbf{y} = \{y_i\}_{i=1}^m$ and $\mathbf{z} = \{z_i\}_{i=1}^m$ are then used as the input data pair for training the GPLVM.

3.3. Training a Shape Model

The GPLVM [6] is known to be an effective approach for probabilistically modeling high dimensional data that lies on a low dimensional non-linear manifold. The motivation of using GPLVM to learn the 3D shape prior is based on our observation that the shapes of the same class of objects are usually controlled by a small number of parameters notwithstanding the complex geometrical or topological structures. Through manipulating these factors, we can then model the 3D shape of the whole class of objects with much fewer values and the detailed reconstruction from the 2D image can be estimated with much less difficulty.

In the setting of the reconstruction problem, the training data includes 2D position features and depth features, which are given in pairs. We aim to recover the underlying low-dimensional sub-manifold structure that can model such pair-wise relationships. To model this relationship, we adopt a shared-GPLVM [10], a variant of GPLVM which handles multiple observations that share the same latent structure. Our version is slightly different from the model presented in [10] since we do not adopt the back-constraints to model this inverse mapping because it can be unsuitable for predicting the multiple hypotheses caused by the ambiguity in the single view reconstruction problem.

In our problem, N pairs of position and depth features: $(\mathbf{Y}, \mathbf{Z}) = [(\mathbf{y_1}, \mathbf{z_1}),$
$(\mathbf{y_2}, \mathbf{z_2}), \cdots, (\mathbf{y}_N, \mathbf{z}_N)]$ obtained from the previous subsection, are given as the training data of the model. In the shared GPLVM, such a manifold structure is described by q-dimensional latent variables $\mathbf{X} =$

[2] More details of $\mathbf{G_y}$ and $\mathbf{G_z}$ can be found in Appendix B.

[3] The number of eigenvectors m can be set differently for $\mathbf{G_y}$ and $\mathbf{G_z}$. We here use the same m for the convenience purpose.

$[\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_N}]$ and each $\mathbf{x_i}$ simultaneously controls the corresponding observed PCA feature pair $(\mathbf{y_i}, \mathbf{z_i})$, $i = 1, 2, \cdots, N$. Observations $\mathbf{Y}$ and $\mathbf{Z}$ are conditional independent given the latent structure $\mathbf{X}$.

With the assumption of dimension independence, the likelihood of observations can be formulated as the following product of m independent Gaussian processes:

$$P(\mathbf{Y}|\mathbf{X}, \theta_\mathbf{Y}) = \prod_{i=1}^{m} \mathcal{N}(\mathbf{Y}_{:,\mathbf{i}}; 0, \mathbf{K_Y}), \tag{8}$$

$$P(\mathbf{Z}|\mathbf{X}, \theta_\mathbf{Z}) = \prod_{i=1}^{m} \mathcal{N}(\mathbf{Z}_{:,\mathbf{i}}; 0, \mathbf{K_Z}), \tag{9}$$

where $\mathcal{N}(*; *, *)$ denotes a Gaussian distribution; $\mathbf{Y}_{:,\mathbf{i}}$ and $\mathbf{Z}_{:,\mathbf{i}}$ denote the $N \times 1$ column vectors constructed from the i-th dimension of $\mathbf{Y}$ and $\mathbf{Z}$, respectively; $\mathbf{K_Y} = [K_Y^{(i,j)}]_{1\le i\le N, 1\le j\le N}$ and $\mathbf{K_Z} = [K_Z^{(i,j)}]_{1\le i\le N, 1\le j\le N}$ are kernel matrices which are defined as "RBF+linear" kernels [9] in this paper:

$$\begin{aligned} K_Y^{(i,j)} &= k_Y(\mathbf{x_i}, \mathbf{x_j}) \\ &= \theta_{Y,1} e^{-\frac{\theta_{Y,2}}{2}(\mathbf{x_i}-\mathbf{x_j})^T(\mathbf{x_i}-\mathbf{x_j})} + \theta_{Y,3}^{-1}\delta_{ij} + \theta_{Y,4}\mathbf{x_i}^T\mathbf{x_j}; \end{aligned} \tag{10}$$

$$\begin{aligned} K_Z^{(i,j)} &= k_Z(\mathbf{x_i}, \mathbf{x_j}) \\ &= \theta_{Z,1} e^{-\frac{\theta_{Z,2}}{2}(\mathbf{x_i}-\mathbf{x_j})^T(\mathbf{x_i}-\mathbf{x_j})} + \theta_{Z,3}^{-1}\delta_{ij} + \theta_{Z,4}\mathbf{x_i}^T\mathbf{x_j}. \end{aligned} \tag{11}$$

where δ_{ij} is the Kronecker delta function; and $\theta_\mathbf{Y} = \{\theta_{Y,i}\}_{i=1}^4$ and $\theta_\mathbf{Z} = \{\theta_{Z,i}\}_{i=1}^4$ refer to the hyper-parameters in $\mathbf{K_Y}$ and $\mathbf{K_Z}$, respectively.

And we assume the prior of the latent variables $\mathbf{X}$ to be the product of independent Gaussian distributions on each latent dimension $\mathbf{x_i}$:

$$P(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n; 0, I). \tag{12}$$

Such a prior formulation can pose a regularity and penalize extremely large latent coordinates.

The model minimizes the negative log joint marginal posterior L with respect to both the latent coordinates $\mathbf{X}$ of the training data and the hyper-parameters $\theta_\mathbf{Y}$ and $\theta_\mathbf{Z}$ of the kernels $\mathbf{K_Y}$ and $\mathbf{K_Z}$, where

$$\begin{aligned} L &= -\log P(\mathbf{X}, \mathbf{Y}, \mathbf{Z}|\theta_\mathbf{Y}, \theta_\mathbf{Z}) \\ &= -\log P(\mathbf{Y}|\mathbf{X}, \theta_\mathbf{Y}) P(\mathbf{Z}|\mathbf{X}, \theta_\mathbf{Z}) P(\mathbf{X}) \\ &= \frac{1}{2} tr(\mathbf{K_Y^{-1}}\mathbf{Y}\mathbf{Y}^T) + \frac{1}{2} tr(\mathbf{K_Z^{-1}}\mathbf{Z}\mathbf{Z}^T) + \frac{1}{2}\|\mathbf{X}\|^2 \\ &+ \frac{m}{2}\log|\mathbf{K_Y}| + \frac{m}{2}\log|\mathbf{K_Z}| + const. \end{aligned} \tag{13}$$

Generally speaking, there is no closed-form solution to the optimization problem in (13) when general nonlinear kernels $\mathbf{K_Y}$ and $\mathbf{K_Z}$ are given and there are likely to be multiple local optima [6]. We use the scaled conjugate gradient (SCG) method [25] to minimize (13) with an Isomap [26] initialization on the latent positions $\mathbf{X}$ and a random initialisation on $\theta_\mathbf{Y}$ and $\theta_\mathbf{Z}$ in our implementation. Derivatives required to compute the gradients of 13) are provided in Appendix A.

3.4. Inferring Depths from the Single Silhouette

A straightforward method to predict the depth from a new single-view silhouette input includes the following steps. First, we obtain a 2D-position feature $\tilde{\mathbf{y}}$ from the input silhouette. This is done by going through the same shape registration and dimension reduction procedures as the training stage does. Second, given the GPLVM shape prior $\mathcal{M}$ learned in Section 3.3, the depth feature $\tilde{\mathbf{z}}$ can be inferred from $\tilde{\mathbf{y}}$. It is theoretically equivalent to maximizing the following conditional distribution.

$$\begin{aligned} P(\tilde{\mathbf{z}}|\tilde{\mathbf{y}}, \mathcal{M}) &= P(\tilde{\mathbf{z}}|\tilde{\mathbf{y}}, \mathbf{Z}, \mathbf{Y}, \mathbf{X}, \theta_\mathbf{Y}, \theta_\mathbf{Z}) \\ &= \int P(\mathbf{x}|\tilde{\mathbf{y}}, \mathbf{Y}, \mathbf{X}, \theta_\mathbf{Y}) P(\tilde{\mathbf{z}}|\mathbf{x}, \mathbf{Z}, \mathbf{X}, \theta_\mathbf{Z}) \mathrm{d}\mathbf{x}, \end{aligned} \tag{14}$$

Since there is no closed-form solution to maximize the integral in (14), we approximate the prediction with a two-stage process. In the first stage, we shall find the position $\tilde{\mathbf{x}}$ in the latent space which is most likely to generate the observed 2D-position feature. Unfortunately, GPLVM does not give a simple functional representation for this inverse mapping. Hence, we find the latent position $\tilde{\mathbf{x}}$ by minimizing the negative log predictive posterior $-\log P(\mathbf{x}|\tilde{\mathbf{y}}, \mathbf{Y}, \mathbf{X}, \theta_\mathbf{Y})$.

$$\begin{aligned} \tilde{\mathbf{x}} &= \mathrm{argmin}_x - \log P(\mathbf{x}|\tilde{\mathbf{y}}, \mathbf{Y}, \mathbf{X}, \theta_\mathbf{Y}) \\ &= \mathrm{argmin}_x - \log P(\tilde{\mathbf{y}}|\mathbf{x}, \mathbf{Y}, \mathbf{X}, \theta_\mathbf{Y}) P(\mathbf{x}) \\ &= \mathrm{argmin}_x \Big(\frac{(\tilde{\mathbf{y}} - \mu_\mathbf{y}(\mathbf{x}))^T (\tilde{\mathbf{y}} - \mu_\mathbf{y}(\mathbf{x}))}{2\sigma_y^2(\mathbf{x})} \\ &+ \frac{m}{2}\log(\sigma_y^2(\mathbf{x})) + \frac{1}{2}\|\mathbf{x}\|^2 \Big). \end{aligned} \tag{15}$$

where

$$\mu_y(\mathbf{x}) = \mathbf{k_Y}(\mathbf{x}, \mathbf{X})^T \mathbf{K_Y}^{-1} \mathbf{Y} \tag{16}$$

$$\sigma_y^2(\mathbf{x}) = k_Y(\mathbf{x}, \mathbf{x}) - \mathbf{k_Y}(\mathbf{x}, \mathbf{X})^T \mathbf{K_Y^{-1}} \mathbf{k_Y}(\mathbf{x}, \mathbf{X}), \tag{17}$$

Intuitively, (15) minimizes the reconstruction error (the first term) while keeping the predicative variance $\tilde{\mathbf{y}}$ small (the second term), i.e., the latent position $\tilde{\mathbf{x}}$ of the new instance is close to those of the training data. The last term is a regularity term that penalizes large latent positions and usually has relatively little influence on the optimization. The scale conjugate gradient (SCG)

method is again used for the optimization in our implementation (see Appendix A for detailed formulations).

Equation (15) can usually be multi-modal, which means that the same 2D-position feature input can correspond to several solutions in the latent space. We hence adopt a multiple-initialization scheme to search multiple peaks.

In the second stage, the depth feature $\tilde{\mathbf{z}}$ is to be estimated based on the latent positions we have found.

$$\tilde{\mathbf{z}} = \text{argmax}_z P(\mathbf{z}|\tilde{\mathbf{x}}, \mathbf{Z}, \mathbf{X}, \theta_{\mathbf{Z}}). \tag{18}$$

The second stage is the forward mapping and it has a Gaussian closed-form representation as follows:

$$P(\mathbf{z}|\tilde{\mathbf{x}}, \mathbf{Z}, \mathbf{X}, \theta_{\mathbf{Z}}) = \mathcal{N}(\mathbf{z}; \mu_{\mathbf{z}}(\tilde{\mathbf{x}}), \sigma_z^2(\tilde{\mathbf{x}})\mathbf{I}), \tag{19}$$

where

$$\mu_{\mathbf{z}}(\tilde{\mathbf{x}}) = \mathbf{k}_{\mathbf{Z}}(\tilde{\mathbf{x}}, \mathbf{X})^T \mathbf{K}_{\mathbf{Z}}^{-1} \mathbf{Z}, \tag{20}$$

$$\sigma_z^2(\tilde{\mathbf{x}}) = k_Z(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) - \mathbf{k}_{\mathbf{Z}}(\tilde{\mathbf{x}}, \mathbf{X})^T \mathbf{K}_{\mathbf{Z}}^{-1} \mathbf{k}_{\mathbf{Z}}(\tilde{\mathbf{x}}, \mathbf{X}) \tag{21}$$

Hence the most probable depth features that correspond to each optimized latent point found in the first stage can be simply given by the mean prediction $\tilde{\mathbf{z}} = \mathbf{k}_{\mathbf{Z}}(\tilde{\mathbf{x}}, \mathbf{X}){\mathbf{K}_{\mathbf{Z}}}^{-1}\mathbf{Z}$. The variance σ_z^2, on the other hand, indicates the confidence level of the prediction, or in the other words, an uncertainty measurement. The larger the variance, the more uncertain is the prediction.

Finally, the complete depth maps of the 3D object can be fully reconstructed from the estimated depth feature $\tilde{\mathbf{z}}$ according to (7) through a linear combination of the mean depth vector $\mathbf{G}_{\mathbf{z},\mathbf{0}}$ and the PCA eigenvectors $\mathbf{B} = [\mathbf{b_i}]_{i=1}^m$ which are stored in memory. It follows that the predictive depth maps are subjected to the following Gaussian distribution:

$$P(\mathbf{G}_{\mathbf{z}}|\tilde{\mathbf{x}}, \mathbf{X}, \mathbf{Z}) = \mathcal{N}(\tilde{\mathbf{x}}; \mathbf{G}_{\mathbf{z},\mathbf{0}} + \mathbf{B}\mu_{\mathbf{z}}(\tilde{\mathbf{x}}), \sigma_z^2(\tilde{\mathbf{x}})\mathbf{B}\mathbf{B}^{\mathbf{T}}), \tag{22}$$

which gives the mean depth-map prediction as well as a measurement of uncertainty. From (22), we can further write the variance of the depth map prediction at each grid point as:

$$\sigma_{\mathbf{G}_{\mathbf{z}},i}^2 = \sigma_z^2(\tilde{\mathbf{x}})\|\mathbf{B}_{\mathbf{i},:}\|^2. \tag{23}$$

This point-wise uncertainty measurement is essential for judging the local accuracy of the prediction.

3.5. Shape-from-silhouettes in the Single-View or Sparse-View Setting

Section 3.4 gave a probabilistic approach for directly predicting the 3D shape from a single silhouette. However there are a couple of shortcomings with this approach. First, it requires that testing instances are given in the similar form as the training data. As the training data are usually aligned (usually in the frontal view) and with little noise, the power of the approach is quite limited. Second, the approach is not straightforwardly extendable to adopt additional inputs (e.g., a silhouette in an extra view) for the purpose of ambiguity removal or more accurate reconstruction.

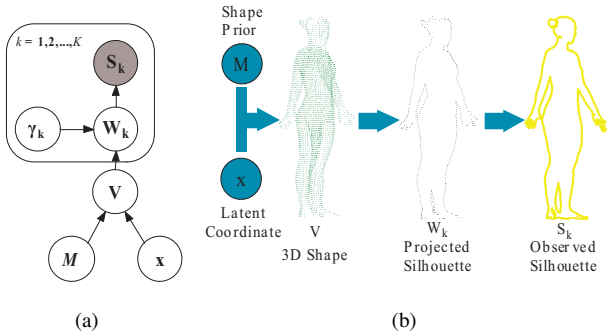


Figure 4: Graphical models of the extended shape prediction framework.

In this subsection, we propose a more adaptive framework for the shape prediction stage. Our aim is to use the same shape model learned in Section 3.3 to handle more complicated issues in the shape inference, e.g., noisy silhouette inputs and outliers, camera viewpoint changes, inputs of more than one view, etc., and achieve more accurate and robust reconstruction.

The problem of modeling 3D shapes from K silhouettes can be described by the graphical model shown in Fig. 4(a). Here we assume that silhouettes are extracted from distinct viewpoints. In the model, $\{\mathbf{S_k}\}_{k=1}^K$ denotes the observed input silhouettes in K cameras, which are given in the form of a 2D point set; $\mathbf{V} = \{\mathbf{v_i}\}_{i=1}^{2N_g}$ represents the set of $2N_g$ 3D sampling points on frontal and dorsal scans generated by the existing shape model $\mathcal{M}$ learned in Section 3.3 (see Appendix B for detailed formulations); and $\mathbf{W_k}$ ($k = 1, 2, \cdots, K$) represents the true silhouette of $\mathbf{V}$ in the k-th view, and it is a 2D point set generated by projecting $\mathbf{V}$ into the k-th camera and extracting those points on the boundary. $\mathbf{V}$, $\mathbf{W_k}$, and $\mathbf{S_k}$ are examplified in Fig. 4(b). Now, the joint posterior can thus be written as:

$$\begin{aligned} &P(\mathbf{V}, \{\mathbf{S_k}, \mathbf{W_k}\}_{k=1}^K | \{\gamma_\mathbf{k}\}_{k=1}^K, \mathcal{M}, \mathbf{x}) \\ =&\Big(\prod_{k=1}^K P(\mathbf{S_k}|\mathbf{W_k}) P(\mathbf{W_k}|\mathbf{V}, \gamma_\mathbf{k})\Big) P(\mathbf{V}|\mathbf{x}, \mathcal{M}). \end{aligned} \tag{24}$$

where $\mathcal{M} = \{\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \theta_\mathbf{Y}, \theta_\mathbf{Z}\}$ refers to the learned GPLVM model; $\mathbf{x}$ is the latent position; $\gamma_\mathbf{k} = \{\mathbf{P_k}, \mathbf{t_k}\}$ are parameters of the k-th camera. Here, we assume an

affine camera given the fact that objects are usually in weak perspective scenes, where $\mathbf{P_k}$ is the 2×3 projection matrix and $\mathbf{t_k}$ is the 2×1 offset vector of the k-th camera, respectively.

In (24), the terms $P(\mathbf{S_k}|\mathbf{W_k})$ and $P(\mathbf{W_k}|\mathbf{V}, \gamma_\mathbf{k})$ model how well 3D shape $\mathbf{V}$ matches the observed silhouettes $\mathbf{S_k}$ ($k = 1, 2, \cdots, K$). It is formulated as a two-stage process in our approach: the projection stage and the matching stage. In the projection stage, $P(\mathbf{W_k}|\mathbf{V}, \gamma_\mathbf{k})$ models the procedure of projecting the 3D shape $\mathbf{V}$ into a silhouette $\mathbf{W_k}$ in the k-th view. It is defined as a Gaussian distribution shown in (25).

$$P(\mathbf{W_k}|\mathbf{V}, \gamma_\mathbf{k}) = \mathcal{N}(\mathbf{W_k}; \tilde{\mathbf{P}}_\mathbf{k}\mathbf{V} + \tilde{\mathbf{t}}_\mathbf{k}, \sigma_w^2 \mathbf{I}_{2N_k^b \times 2N_k^b}). \quad (25)$$

where $\tilde{\mathbf{P}}_\mathbf{k} = \mathbf{P_k} \otimes \mathbf{M_k}$ and $\tilde{\mathbf{t}}_\mathbf{k} = \mathbf{t_k} \otimes \mathbf{1}_{\mathbf{N_k^b} \times \mathbf{1}}$ are the expanded versions of the projection matrix $\mathbf{P_k}$ and the offset vector $\mathbf{t_k}$ in the k-th view, respectively. Here, $\mathbf{V}$ is represented by a $6N_g$-D column vector concatenated by all $2N_g$ 3D sampling points, and $\mathbf{W_k}$ is represented by a $2N_k^b$-D column vector concatenated by the 2D image positions of the N_k^b sample points on the boundary in the k-th view; $\mathbf{M_k} = [m_{k,ij}]_{1 \le i \le N_k^b, 1 \le j \le 2N_g}$ is a $N_k^b \times 2N_g$ binary masking matrix with element $m_{k,ij} = 1$ if the projection of the i-th 3D sample points is on the boundary and $m_{k,ij} = 0$ otherwise. $\mathbf{M_k}$ selects the N_k^b silhouette points of the projection in the k-th view. Both $\mathbf{M_k}$ and N_k^b are fully determined by $\mathbf{P_k}$.

In the matching stage, $P(\mathbf{S_k}|\mathbf{W_k})$ models how well the input silhouette $\mathbf{S_k}$ fits the corresponding boundary projection $\mathbf{W_k}$ of the generated shape in the k-th view. As shown in (26), the observation likelihood is defined on the basis of Chamfer matching, which is robust to errors and outliers in the input silhouettes and has been widely used for matching silhouettes in the application of object recognition and shape recovery [5], [20], [27].

$$P(\mathbf{S_k}|\mathbf{W_k}) = \frac{1}{Z} \exp\Big(-\frac{1}{2\sigma_s^2} DT^2_{\mathbf{S_k}}(\mathbf{W_k})) \Big), \quad (26)$$

where $DT^2_{\mathbf{S_k}}(\cdot)$ refers to the squared L2-distance transform of the silhouette $\mathbf{S_k}$. For an arbitrary point set $\mathbf{W} = \{\mathbf{w_1}, \mathbf{w_2}, \cdots, \mathbf{w_n}\}$, it is defined as $DT^2_{\mathbf{S_k}}(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^n \min_{\mathbf{u_i} \in \mathbf{S_k}} \|\mathbf{w_i} - \mathbf{u_i}\|^2$. To simplify the computation, the normalization factor Z is approximated by a constant here.

Finally, the last term in (24) generates the 3D shape $\mathbf{V}$ from the learned shape model $\mathcal{M}$ at the latent position $\mathbf{x}$, which is formulated as the predictive likelihood of the learned GPLVM:

$$P(\mathbf{V}|\mathbf{x}, \mathcal{M}) = \mathcal{N}(\mathbf{V}; \mu_\mathbf{V}(\mathbf{x}), \mathbf{\Sigma_V}(\mathbf{x})). \quad (27)$$

And it can be further shown that the likelihood $P(\mathbf{W_k}|\mathbf{x}, \mathcal{M}, \gamma_\mathbf{k})$ also has the Gaussian form:

$$\begin{aligned} P(\mathbf{W_k}|\mathbf{x}, \mathcal{M}, \gamma_\mathbf{k}) &= \int_\mathbf{V} P(\mathbf{W_k}|\mathbf{V}, \gamma_\mathbf{k}) P(\mathbf{V}|\mathbf{x}, \mathcal{M}) d\mathbf{V} \\ &= \mathcal{N}(\mathbf{W_k}; \mu_{\mathbf{W_k}}(\mathbf{x}, \gamma_\mathbf{k}), \mathbf{\Sigma}_{\mathbf{W_k}}(\mathbf{x}, \gamma_\mathbf{k})). \end{aligned} \quad (28)$$

The detailed formulations of $\mu_\mathbf{V}, \mathbf{\Sigma_V}, \mu_{\mathbf{W_k}}, \mathbf{\Sigma}_{\mathbf{W_k}}$ [4] as well as related derivations are given in Appendix B.

Our target is to find the optimal 3D shape that best fits all the image evidences $\mathbf{S_k}$ ($k = 1, 2, \cdots, K$) in K view, or equivalently, to find the latent position $\mathbf{x}$. And on the other hand, we also hope to correct parameters $\gamma_\mathbf{k}$ of K cameras. This can be done by finding the maximum of the overall likelihood of $\mathbf{S_k}$ given $\mathbf{x}$ and $\gamma_\mathbf{k}$ ($k = 1, 2, \cdots, K$) as follows.

$$\begin{aligned} &P(\{\mathbf{S_k}\}_{k=1}^K|\mathbf{x}, \mathcal{M}, \{\gamma_\mathbf{k}\}_{k=1}^K) = \prod_{k=1}^K P(\mathbf{S_k}|\mathbf{x}, \mathcal{M}, \gamma_\mathbf{k}) \\ &= \prod_{k=1}^K \int_{\mathbf{W_k}} P(\mathbf{S_k}|\mathbf{W_k}) P(\mathbf{W_k}|\mathbf{x}, \mathcal{M}, \gamma_\mathbf{k}) d\mathbf{W_k} \\ &= \prod_{k=1}^K \int_{\mathbf{W_k}} \frac{1}{Z_k} e^{-\frac{1}{2\sigma_s^2} DT^2_{\mathbf{S_k}}(\mathbf{W_k})} \mathcal{N}(\mathbf{W_k}; \mu_{\mathbf{W_k}}, \mathbf{\Sigma}_{\mathbf{W_k}}) d\mathbf{W_k}, \end{aligned} \quad (29)$$

The likelihood has no closed form since the direct integral over the terms with distance transform is not tractable. As a consequence, a direct maximization on (29) will be computationally troublesome. However, the following property of the L2-distance transform will help the computation.

Property 1. *Let* $\mathbf{S}$ *be the observed 2D silhouette, and* $\mathbf{W} = \{\mathbf{w_1}, \mathbf{w_2}, \cdots, \mathbf{w_n}\}$ *and* $\mathbf{U} = \{\mathbf{u_1}, \mathbf{u_2}, \cdots, \mathbf{u_n}\}$ *be two 2D point sets of size n. Then the squared L2-distance transform of* $\mathbf{W}$ *satisfies* $DT^2_\mathbf{S}(\mathbf{W}) + DT^2_\mathbf{S}(\mathbf{U}) \le 2DT^2_\mathbf{S}(\frac{\mathbf{U}+\mathbf{W}}{2}) + \frac{1}{2n}\|\mathbf{W} - \mathbf{U}\|^2$.

Proof.

$$\begin{aligned} l.h.s &= \frac{1}{n}\sum_{i=1}^n \min_{\mathbf{v_i} \in \mathbf{S}} \|\mathbf{w_i} - \mathbf{v_i}\|^2 + \min_{\mathbf{v_i'} \in \mathbf{S}} \|\mathbf{u_i} - \mathbf{v_i'}\|^2 \\ &\le \frac{1}{n}\sum_{i=1}^n \min_{\mathbf{v_i} \in \mathbf{S}} \left(\|\mathbf{w_i} - \mathbf{v_i}\|^2 + \|\mathbf{u_i} - \mathbf{v_i}\|^2\right) \end{aligned}$$

[4] For the convenience of notation, we sometimes omit the parameters of the terms, e.g., $\mu_\mathbf{V} = \mu_\mathbf{V}(\mathbf{x})$.

$$= \frac{1}{n}\sum_{i=1}^{n}\min_{\mathbf{v_i}\in\mathbf{S}}(2\|\frac{\mathbf{w_i}+\mathbf{u_i}}{2}-\mathbf{v_i}\|^2+2\|\frac{\mathbf{w_i}-\mathbf{u_i}}{2}\|^2)$$
$$= \frac{2}{n}\sum_{i=1}^{n}\min_{\mathbf{v_i}\in\mathbf{S}}\|\frac{\mathbf{w_i}+\mathbf{u_i}}{2}-\mathbf{v_i}\|^2+\frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{w_i}-\mathbf{u_i}\|^2$$
$$= r.h.s.$$

□

Property 1 give a closed-form lower bound $Q(\mathbf{x},\{\gamma_\mathbf{k}\}_{k=1}^K)$ to the likelihood $P(\{\mathbf{S_k}\}_{k=1}^K|\mathbf{x},\mathcal{M},\{\gamma_\mathbf{k}\}_{k=1}^K)$, as shown in (30).

$$\begin{aligned}
&P(\{\mathbf{S_k}\}_{k=1}^K|\mathbf{x},\mathcal{M},\{\gamma_\mathbf{k}\}_{k=1}^K)\\
=&\prod_{k=1}^K\int_{\mathbf{W_k}}\frac{1}{2Z_k}(e^{-\frac{1}{2\sigma_s^2}DT^2_{\mathbf{S_k}}(\mathbf{W_k})}+e^{-\frac{1}{2\sigma_s^2}DT^2_{\mathbf{S_k}}(2\mu_{W_k}-\mathbf{W_k})})\\
&\cdot\mathcal{N}(\mathbf{W_k};\mu_{\mathbf{W_k}},\mathbf{\Sigma}_{\mathbf{W_k}})d\mathbf{W_k}\\
\geq&\prod_{k=1}^K\int_{\mathbf{W_k}}\frac{1}{2Z_k}(e^{-\frac{1}{4\sigma_s^2}\left(DT^2_{\mathbf{S_k}}(\mathbf{W_k})+DT^2_{\mathbf{S_k}}(2\mu_{W_k}-\mathbf{W_k})\right)})\\
&\cdot\mathcal{N}(\mathbf{W_k};\mu_{\mathbf{W_k}},\mathbf{\Sigma}_{\mathbf{W_k}})d\mathbf{W_k}\\
\geq&\prod_{k=1}^K\frac{1}{Z_k}\exp(-\frac{1}{2\sigma_s^2}DT^2_{\mathbf{S_k}}(\mu_{\mathbf{W_k}}))\\
&\cdot\prod_{k=1}^K\int_{\mathbf{W_k}}e^{-\frac{1}{2N_k^b\sigma_s^2}\|\mathbf{W_k}-\mu_{\mathbf{W_k}}\|^2}\mathcal{N}(\mathbf{W_k};\mu_{\mathbf{W_k}},\mathbf{\Sigma}_{\mathbf{W_k}})d\mathbf{W_k}\\
=&\prod_{k=1}^K\frac{1}{Z_k\sqrt{\det(\mathbf{I}+\frac{1}{N_k^b\sigma_s^2}\mathbf{\Sigma}_{\mathbf{W_k}})}}\exp\Big(-\frac{1}{2\sigma_s^2}DT^2_{\mathbf{S_k}}(\mu_{\mathbf{W_k}})\Big)\\
=&Q(\mathbf{x},\{\gamma_\mathbf{k}\}_{k=1}^K).
\end{aligned}\tag{30}$$

Practically, the maximum-likelihood estimate of the latent coordinate $\mathbf{x}_{ML}$ and camera parameters $\gamma_{\mathbf{k}ML}$ ($k=1,2,\cdots,K$) can be well approximated by finding $\mathbf{x}$ and $\{\gamma_\mathbf{k}\}_{k=1}^K$ which maximize the lower bound Q.

$$\begin{aligned}
(\mathbf{x}_{ML},\{\gamma_{\mathbf{k}ML}\}_{k=1}^K)&\approx\arg\max_{\mathbf{x},\{\gamma_\mathbf{k}\}_{k=1}^K}Q(\mathbf{x},\{\gamma_\mathbf{k}\}_{k=1}^K)\\
&=\arg\min_{\mathbf{x},\{\gamma_\mathbf{k}\}_{k=1}^K}-\log Q(\mathbf{x},\{\gamma_\mathbf{k}\}_{k=1}^K)\\
&=\arg\min_{\mathbf{x},\{\gamma_\mathbf{k}\}_{k=1}^K}\sum_{k=1}^K\Big(\frac{1}{2\sigma_s^2}DT^2_{\mathbf{S_k}}(\mu_{\mathbf{W_k}})\\
&\quad+\frac{1}{2}\log\det(\mathbf{I}+\frac{1}{N_k^b\sigma_s^2}\mathbf{\Sigma}_{\mathbf{W_k}})\Big),
\end{aligned}\tag{31}$$

and consequently the corresponding maximum likelihood estimate of the 3D shape can be given as:

$$P(\mathbf{V}_{ML}|\mathbf{x}_{ML},\mathcal{M})=\mathcal{N}(\mathbf{V}_{ML};\mu_\mathbf{V}(\mathbf{x}_{ML}),\mathbf{\Sigma}_\mathbf{V}(\mathbf{x}_{ML})),\tag{32}$$

In our implementation, $-\log Q$ is optimized by the adaptive-scale line search and multiple initializations are used to avoid local minima. The optimisation alternates between finding the latent coordinate $\mathbf{x}$ and correcting camera parameters $\{\gamma_\mathbf{k}\}_{k=1}^K$ (and hence $\{\mathbf{M_k}\}_{k=1}^K$ and $\{N_k^b\}_{k=1}^K$). The convergence usually comes fast, as the latent dimension of GPLVM is low and a small perturbation of the camera parameters is assumed. It is also worth to mention that the determinant $\det(\mathbf{I}+\frac{1}{N_k^b\sigma_s^2}\mathbf{\Sigma}_{\mathbf{W_k}})$ in (31) can be computed efficiently (see Appendix C).

3.6. Issues on Computational Complexity

The computational complexity of the GPLVM is mainly dependent on the number of training data. Let N be the size of the training set, and m be dimensionality of the observation data (position and depth features in our problem). The computational complexity of GPLVM training is determined by the inversion of kernel matrices, i.e., $O(N^3)$; while in the depth prediction stage, the complexity is $O(mN)$ for each iteration, which is determined by matrices multiplications in (15) and (18), where $\mathbf{K_Z^{-1}Z}$ can be calculated off-line and stored.

A sparse method, named the Informative Vector Machine (IVM) [29] can be used to further speed up the training and prediction. In IVM, a small subset of the training data called the *"active set"* (size $d \ll N$) is selected to construct the GPLVM effectively. In the training process, data points are added to the model one at a time, and at each step the point with the highest construction variances (see (17) and (21)) will be selected. In this way, the active set tends to contain those training data points that are reasonably well spaced throughout the latent space. The overall complexity of the training can be reduced to $O(d^2N)$, which is dominated by the active selection, while the complexity of each-step prediction drops to $O(md)$.

4. Experimental Results

4.1. Datasets

In order to verify the efficacy of our approach, we train the shape models on both synthetic and real data.

We first use parametric models to synthesize two 3D shape datasets: a vase dataset (2000 instances) and a mug dataset (2000 instances). The side silhouettes are parametrized by following equations:

$$r_{mug}(h)=r_1h+r_2(1-h)+\frac{f_h}{2}\sqrt{max(r_h^2-(h-0.5)^2,0)}\tag{33}$$

$$\begin{aligned}
&r_{vase}(h)=r_1h+r_2(1-h)+s_1\sin(\pi h^{s_2}),\\
&\qquad\text{where } h\in[0,1],
\end{aligned}\tag{34}$$

Table 1: The parameter ranges of synthetic objects. Suppose that the height of the object is 1.

Dataset	Parameters				
	r_1	r_2	r_h	s_1	s_2
Mugs	$U(0.25, 0.4)$	$U(0.15, 0.25)$	$U(\frac{r_1}{4}, \frac{r_1}{4} + 0.03)$	N/A	N/A
Vases	$U(0.25, 0.4)$	$U(0.15, 0.25)$	N/A	$U(-5, 5)$	$U(0.5, 1.5)$

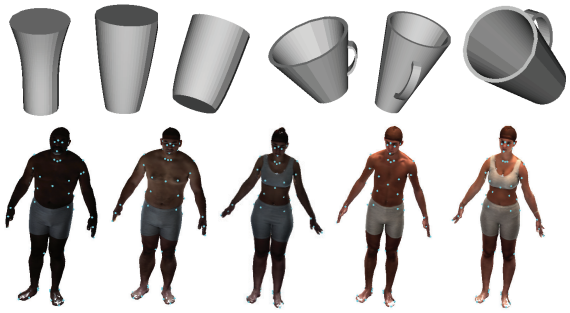


Figure 5: Instances of the datasets we use. Vases (Row 1 left), mugs (Row 1 right), and the human bodies in CAESAR dataset(Row 2. Landmarks are indicated by blue dots.).

where r_1 and r_2 are radii of the top and the bottom, respectively; s_1 and s_2 are parameters for generating curved silhouettes of vases; r_h is the handle size of mugs; and f_h is a binary variable which equals 1 at the handle areas and 0 otherwise. We generate all parameters using uniform distributions with the ranges listed in Table 1. For the real data, we investigate the SAE International Civilian American and European Surface Anthropometry Resource (CAESAR) dataset, which contains over 2000 3D body scans of North American and European adults. For each instance in the CAESAR dataset, 74 anthropometric landmarks are provided and they can be used for registration. Some instances of each dataset are illustrated in Fig. 5.

For registering shapes, the template grid resolutions N_g of the three classes are 760, 760, and 1864, respectively. Concerning the scale ambiguity of the reconstruction problem, all objects in a dataset are normalized to have the same height 1 before the training and the testing.

4.2. Selection of Latent Dimensions

The latent dimension q determines the model complexity of GPLVM and choosing a proper q must be addressed. The latent space may not be enough to characterize the inherent manifold structure of the data when its dimensionality is set too low, while on the other hand, a larger value of q will also increase the average prediction errors because it leads to over-fitting and poor generalization.

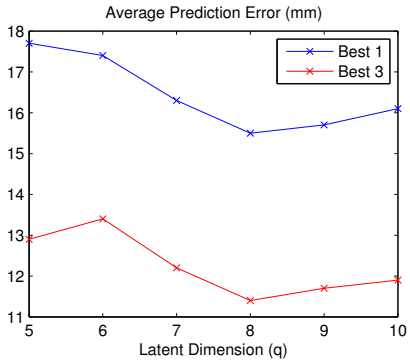
In practice, we can obtain the proper latent dimension q of a dataset $\mathcal{D}$ through cross validations. During the model training stage, we train N_q GPLVMs $\mathcal{M}_1, \mathcal{M}_2, \cdots \mathcal{M}_{N_q}$ on the same subset $\mathcal{D}_1 \subset \mathcal{D}$ with a range of different latent dimensions $q_1, q_2, \cdots, q_{N_q}$, and use the rest data $\mathcal{D} \backslash \mathcal{D}_1$ as the testing set. The model best fitting the testing set will be selected.

In our work, we learn models for each dataset with the latent dimension q ranging from 3 to 10, and use the cross-validation approach described above to validate the performance of these models. The criterion for the model comparison is the average prediction error measured in terms of the mean vertex-to-vertex distance between the model prediction and the ground truth (see Fig. 6). Through experiments, we finally adopt the optimal dimensionality $q = 4$ for the vase data, $q = 4$ for the mug data, and $q = 8$ for the human-body data regardless of gender, respectively.

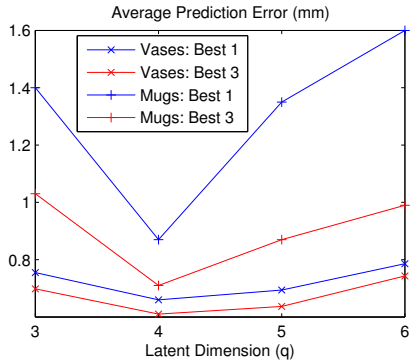
4.3. Qualitative Experiments

We first train the model of each type of shapes using 800 instances of the same category, and then apply our single-silhouette-based prediction approach proposed in Section 3.4 to reconstruct new shape from the testing silhouettes. Some qualitative results are given in Fig. 7. For each testing instance, the model automatically returns several depth distributions. In Fig. 7, we give shapes corresponding to highest predictive posterior values in comparison with the ground truth. Uncertainty measurements of the shapes are also visually provided. In general, the shape model we learn can generate satisfactory reconstruction results. It can be observed that the rotational symmetry of both catagories is well captured. It is also worth mentioning that uncertainty measurements give an important cues on the quality of the local surface estimation. More discussion on this effect will be given in Section 5.

For the mug dataset, we launch two individual experiments by providing training and testing silhouettes in



(a) Human bodies



(b) Vases and Mugs

Figure 6: The comparison of models with different latent dimensions. We plot prediction errors versus the latent dimension of the models.

different settings. In the first setting, the handle of the mug is in the side and not occluded, but it creates an extra loop in the topology of the silhouette. In the other test, the viewpoint is parallel to the handle direction, which may result in the occlusion of the mug handle. In this case, we also deliberately change the direction of mug handle randomly, which can be either in front or behind. Since changing handle position does not affect the external silhouette of the mug, we expect to see a bimodal structure in the GPLVM prior model which is learned from such a dataset. As expected, our model is capable of generating both feasible reconstruction results. Given an ambiguous frontal silhouette of the mug (a trapezoid), our approach is able to find both possible solutions from the two peaks in the predictive posterior (15), each corresponding to the two different handle directions, as shown in Fig. 8.

We also train GPLVM body shape priors on male instances, female instances, and mixed-gender instances, respectively. Each model is learned from the 2D projections and depth maps of 800 instances in standing pose. In the testing stage, we provide only the 2D standing pose silhouettes in frontal view as inputs to the shape model we obtain. The reconstruction results of human bodies with different body shapes and genders are given in Fig. 9. The GPLVM model captures the shape variation among different people, and it is able to generate multiple reasonable candidate body shapes which correspond to the input silhouette.

4.4. Quantitative Evaluations and Comparisons

We also evaluate the accuracy of the single-silhouette reconstruction method quantitatively. For the comparison purpose, we implement two different reconstruction methods in addition to our approach. The first one is

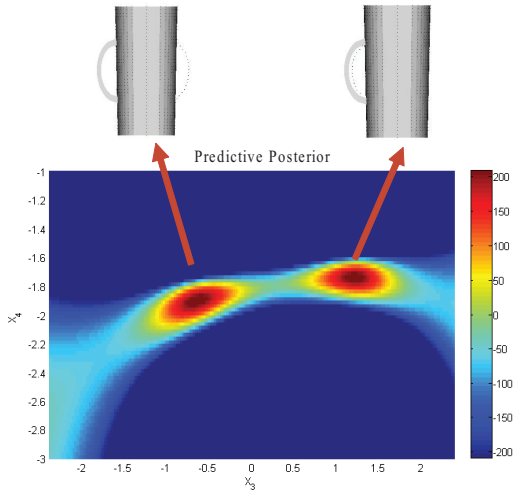


Figure 8: The bimodal predictive posterior of the last mug sample in Fig. 7. Here we fix the first two dimensions of the latent space and plot the distribution of the posterior with respect to the 3rd and 4th dimensions. Each peak of the predictive posterior corresponds to the case of two different handle directions.

based on nearest-neighbors (NN) searching, i.e. finding the instance in the database which gives the most similar silhouette as the query and return its corresponding depth maps. For the second comparative method, we perform PCA on the database such that grid positions and depth maps are encoded jointly. The reconstruction is done by searching the eigen-space and combining the modes such that the generated shape fits the query silhouette. The first 20 principle components are used to capture the variation of the shapes in our implementation. This PCA-based shape modeling approach and similar variants are widely-used in previous literatures [3], [4], [5], [20]. In the implementation of both meth-

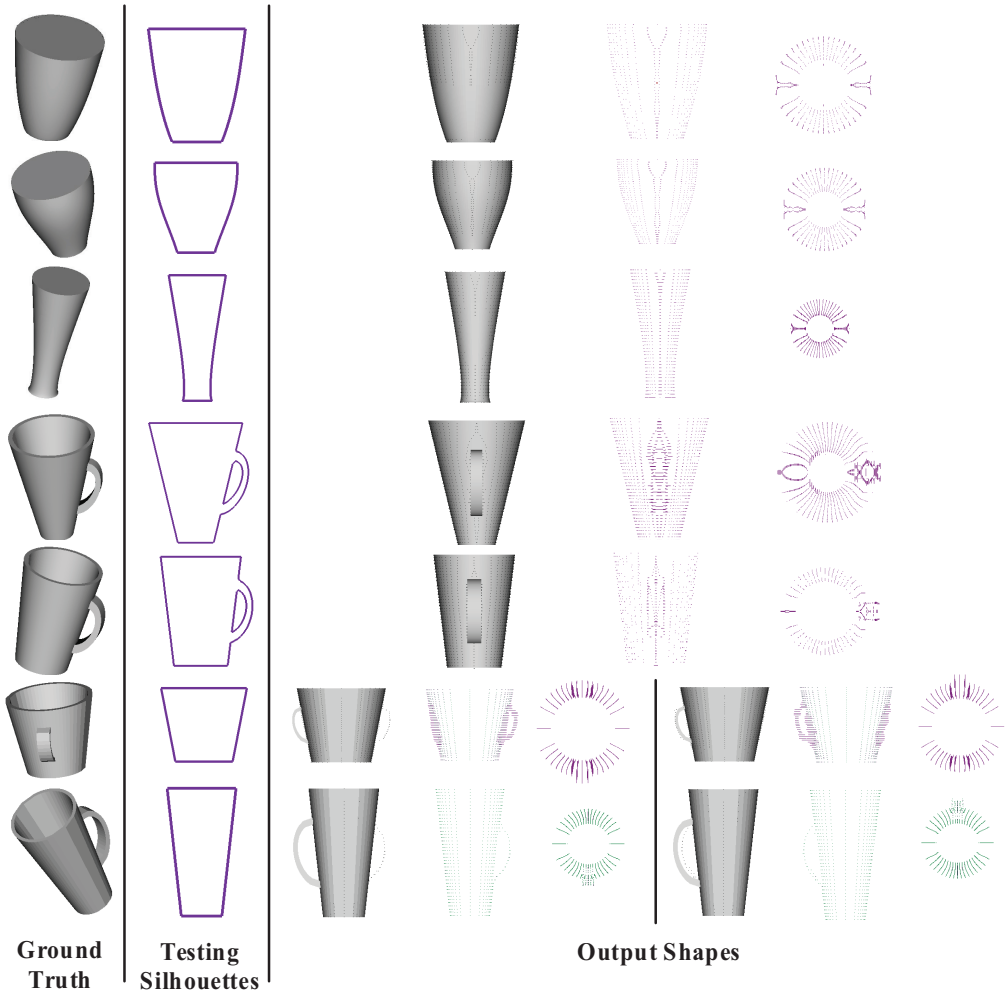


Figure 7: Qualitative results on synthetic mug and vase datasets. Row 1-3: Test on vase silhouettes; Row 4-5: Test on side-view mug silhouettes; Row 6-7: Test on frontal-view mug silhouettes. For each testing instance, we give the shape prediction (in black dots) in contrast with the ground truth in the side view, as well as uncertainty measurement (in magenta error bars $[\mu - \sigma, \mu + \sigma]$) from both the top view and the side view. For the frontal-view mug instances (row 3 and 4), two different modes of the solution are given.

ods, we adopt Hausdorff Fraction [28] as the measurement of silhouette similarity.

We compare the performance of these algorithms on all three datasets. For each dataset, we use 800 instances randomly selected from the datasets to train the GPLVM shape prior and another 800 as the testing set. The exact same training sets and testing sets are used in the nearest-neighbor search and PCA-based reconstruction. The evaluation is based on measuring the errors between the ground truths and the candidate reconstructed results at specified positions. To obtain the natual scales of measurement errors, we assume different actual height H for each class of objects: H = 100mm for vases and mugs, and H = 1700mm for human bodies.

In this paper, the error measurement of the reconstruction is defined as:

$$Err = \frac{1}{N_S} \sum_{i=1}^{N_S} |d_i - d_{i,0}|, \tag{35}$$

where d_i and $d_{i,0}$ are the reconstructed and ground truth thickness values (the difference of frontal and dorsal depth values) at sampling position i, respectively, and N_S is the total number of sampling positions. For the synthetic data, we sample thickness values at all the grid points. On the other hand, for the human body data, thickness values are sampled around the chest and the waist, which are of most interests in anthropometric

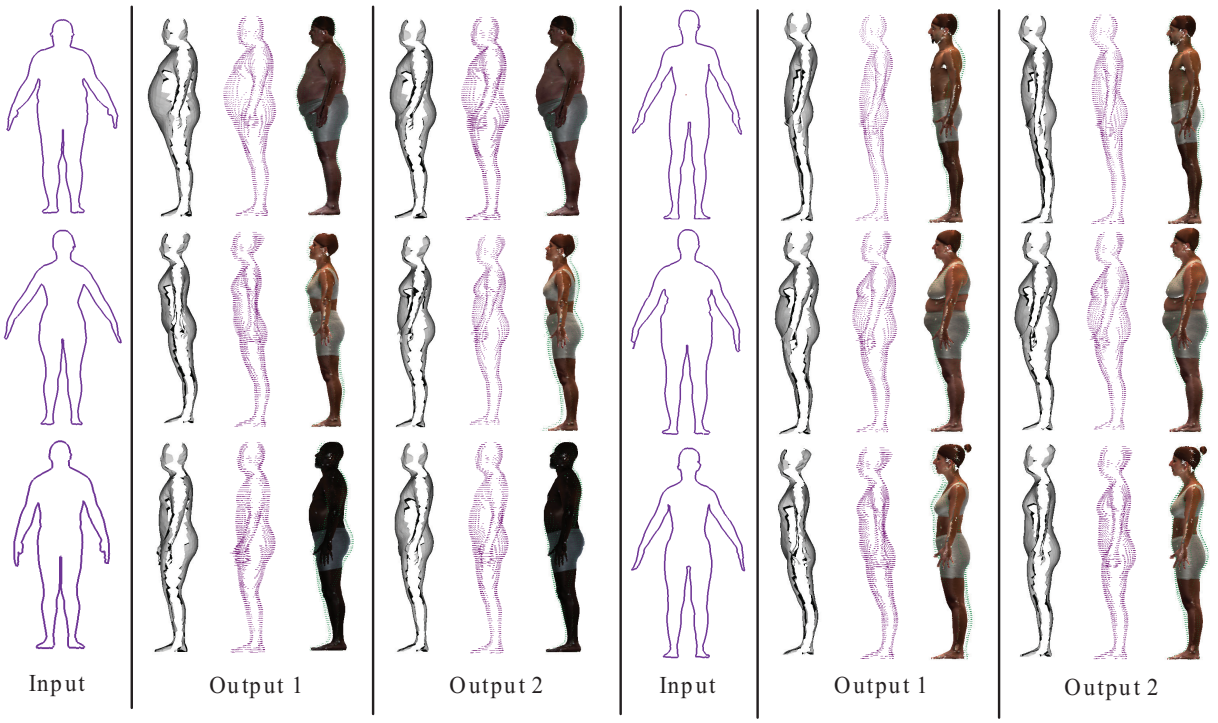


Figure 9: Reconstruction from a single frontal view silhouette of the human body. For each given testing input silhouette (column 1), we predict the depth maps using the trained model. Two candidate reconstruction results with the highest predictive posterior values are given in column 2 and 3. For each result, we give the reconstructed depth surface from the side view (left), the uncertainty of the prediction (center, in magenta error bars $[\mu - \sigma, \mu + \sigma]$), and the contrast with the ground truth (right).

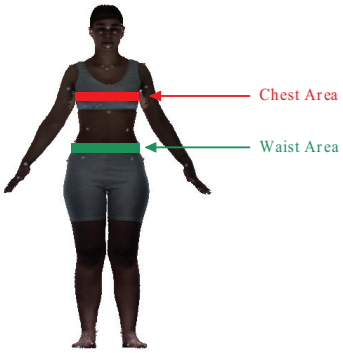


Figure 10: An illustration of chest and waist areas on the human body for depth samplings and error measurements.

measurements. In our experiments, the chest area and the waist area are defined as 5cm-wide horizontal bands (see Fig. 10), and their heights are determined by CAESAR landmark marks "Right/Left Thelion/Bustpoint" and "Waist Preferred, Posterier", respectively.

From the results tabulated in Table 2 and 3, we can see that both our GPLVM-based approach and the PCA-based approach always outperform the nearest-neighbor-based approach in all the contexts listed. In more cases, our approach gives lower predictive errors than the PCA-based approach. This is mainly because searching in a lower dimensional latent space is less likely to be trapped into local optima.

We also compare the efficiency of different approaches, and investigate how the size of the training set affects the speed and the precision of the GPLVM model. We run all the codes on a 2.5GHz processor. The average training and prediction time for different approaches are summarized in Table 4. Generally speaking, the nearest-neighbor method is most efficient due to its simplicity. Our GPLVM-based approach is more efficient than the PCA-based approach due to the fact that the optimization of GPLVM is done on a more compact latent space (usually $\leq$ 8-D), compared with higher-dimensional 20 – 30-D PCA eigen-space.

In the experiments, we train three GPLVMs with the same latent dimension for each dataset based on different training sets: a full training set of 800 instances (GPLVM-F), a subset of the full training set containing only 200 instances selected by random (GPLVM-R), and another subset of 200 instances selected by IVM

Table 2: Quantitative comparisons on vases and mugs datasets given the single frontal-view silhouette. Errors and standard deviations are given in millimeters, and the actual height of each instance is assumed to be 100mm.

Data Set		The 1st Candidate				
		GPLVM	GPLVM-R	GPLVM-I	PCA	NN
Dimensionality		4	4	4	10	
Vases		**0.66 ± 0.24**	0.89 ± 0.37	0.72 ± 0.76	1.08 ± 0.96	2.17 ± 1.76
Mugs	Frontal	0.87 ± 0.40	**0.82 ± 0.49**	0.83 ± 0.47	1.09 ± 0.77	1.60 ± 1.22
	Side	1.19 ± 0.9	1.45 ± 1.19	1.34 ± 1.17	**1.07 ± 0.87**	2.75 ± 2.19
Data Set		Best among the First 3				
		GPLVM	GPLVM-R	GPLVM-I	PCA	NN
Dimensionality		4	4	4	10	
Vases		**0.61 ± 0.29**	0.76 ± 0.2	0.67 ± 0.62	0.94 ± 0.93	1.70 ± 1.12
Mugs	Frontal	**0.71 ± 0.40**	0.76 ± 0.43	0.75 ± 0.45	0.84 ±0.68	1.10 ± 0.65
	Side	1.02 ± 0.92	1.22 ± 1.18	1.13 ± 1.06	**0.98 ± 0.95**	2.37 ± 1.84

Table 4: Empirical time (in seconds) for training and prediction (per instance) for different approaches. For GPLVM and PCA approaches, the prediction includes 20 restarts for finding optima.

Approaches	Training	Prediction
GPLVM ($q = 4, N = 200$)	86	15
GPLVM ($q = 8, N = 200$)	190	43
GPLVM ($q = 4, N = 800$)	2844	224
GPLVM ($q = 8, N = 800$)	5336	877
PCA ($m = 10$)	–	703
PCA ($m = 20$)	–	1898
NN	–	3

(GPLVM-I). Reducing in the number of training samples from 800 to 200 greatly enhances the training and prediction speed. The model training is sped-up by about 30 times, while the average time for depth prediction on each instance is sped-up by about 15 times.

The testing errors in Table 2 and 3 show that the performance of our GPLVM is very slightly affected after the reduction of the training set size. Also, comparing two different data selection schemes, IVM better maintains the performance of the model than a random selection.

4.5. Reconstruction using the Extended Prediction Framework

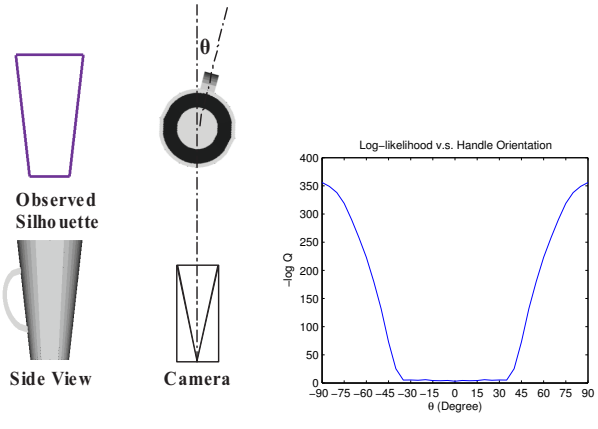
We also carry out qualitative and quantitative experiments on the extended shape prediction approach proposed in Section 3.5. Using the shape priors learned in Section 3.3, the framework can reconstruct 3D shapes from either a single silhouette or multiple silhouettes with different camera viewpoints. In our experiments, we set variance parameters σ_s^2 and σ_w^2 to be 10^{-5} and 10^{-7}, respectively.

First, we revisit the occlusion problem in the mug reconstruction. In Section 4.3, the probability distribution of the handle direction is bimodal given the assumption that the camera viewpoint is in parallel with the handle orientation. We now consider the more general case, the handle direction under an arbitrary camera viewpoint. Given the frontal-view silhouette in Fig. 11(a), we use the approach in Section 3.5 to reconstruct the 3D shape, and plot the negative logarithm of the likelihood lower bound $-\log Q$ against the camera angle parameter θ in Fig. 11(b) with the latent coordinate $\mathbf{x}$ fixed. In the diagram, there is a wide flat region $\theta \in [-35, 35]$ of low $-\log Q$ value, which indicates that all viewpoints which result in the handle occlusion are approximately equi-probable. This matches our perception that the orientation of the occluded handle is completely uncertain unless some prior knowledge on the camera parameter is given. In practice, we are usally much more interested in the shape variation rather than such uncertainty in the viewpoint. Hence, to speed up the shape construction in the following experiments, we intentionally remove this uncertainty by roughly initializing camera projection matrices $\mathbf{P_k}$ and narrowing down the searching range of camera parameters.

We then test the exteneded approach on the human body dataset under a 2-view ($K = 2$) sparse-view setting. More specially, a side-view silhouette is provided in addition to the frontal-view silhouette. The experiments are based on the same training and testing sets as those in Section 4.3. Some of the qualita-

Table 3: Quantitative comparisons on human data given the single frontal-view silhouette. Errors and standard deviations are given in millimeters, and the actual body height of each instance is assumed to be 1700mm.

Data Set		The 1st Candidate				
		GPLVM	GPLVM-R	GPLVM-I	PCA	NN
Dimensionality		8	8	8	20	
Humans (mixed)	Chest	**15.8 ± 13.3**	17.3 ± 13.4	16.5 ± 13.1	21.4 ± 16.7	22.1 ± 16.5
	Waist	**15.6 ± 12.2**	16.5 ± 10.4	**15.6 ± 10.0**	19.0 ± 11.7	19.4 ± 13.8
Humans (female)	Chest	**15.4 ± 12.9**	17.5 ± 13.0	16.5 ± 10.9	17.2 ± 12.4	20.5 ± 15.5
	Waist	15.6 ± 11.7	15.3 ± 12.2	15.5 ± 9.9	**15.0 ± 8.4**	20.0 ± 14.0
Humans (male)	Chest	15.2 ± 13.8	15.6 ± 11.9	**15.1 ± 12.3**	16.6 ± 11.6	20.5 ± 14.2
	Waist	**15.8 ± 11.7**	16.6 ± 11.2	16.8 ± 11.4	17.7 ± 10.5	18.9 ± 11.7
Data Set		Best among the First 3				
		GPLVM	GPLVM-R	GPLVM-I	PCA	NN
Dimensionality		8	8	8	20	
Humans (mixed)	Chest	**10.2 ± 7.3**	12.1 ± 10.5	10.9 ± 9.1	16.2 ± 14.3	12.9 ± 10.7
	Waist	**12.6 ± 7.5**	14.4 ± 7.7	12.9 ± 7.5	17.3 ± 11.2	15.0 ± 10.2
Humans (female)	Chest	**10.6 ± 8.7**	12.0 ± 9.1	10.6 ± 8.1	13.4 ± 9.9	12.7 ± 11.2
	Waist	**12.0 ± 8.4**	13.0 ± 8.3	12.7 ± 7.9	13.9 ± 7.7	14.7 ± 9.9
Humans (male)	Chest	11.4 ± 9.6	11.4 ± 9.3	**11.0 ± 7.4**	13.1 ± 9.8	12.3 ± 8.4
	Waist	**12.3 ± 8.2**	13.3 ± 7.9	12.8 ± 8.2	14.7 ± 8.8	14.9 ± 8.2



(a) (b)

Figure 11: Possible camera poses given the silhouette of a mug with its handle occluded.

tive results from frontal-side silhouette pairs are given in Fig. 12. We compare the results with those reconstructed from the single frontal silhouette only. As expected, we can see that the reconstructed body shape better fits the ground truth shape of the query instances than the single-view result does. Also, it is worth mentioning that the secondary side silhouette helps disambiguate the slight pose changes perpendicular to the image plane, such as leaning forward or backward, which are almost unobservable from the frontal silhouette.

In Table 5, we also provide the quantitative performance of our extended framework in comparison with other approaches. In the case when only the single frontal silhouette is used as the input, the extended framework performs similarly to the straightforward framework in Section 3.4. Benefiting from the additional side-view silhouette, prediction errors of the GPLVM drop as much as $30-50\%$ when comparing with the reconstruction from a single frontal-view silhouette. Compared with the PCA-approach, our approach is able to give similar accuracy but with a much lower dimensionality, which implies a faster optimisation.

Finally, we test our framework on the real images with common background scenes. Segmenting foreground objects from the background with shadows and scenes clutters usually causes additional errors and outliers in the extracted silhouettes, which results in considerable difficulty to the subsequent reconstruction. In our implementation, we adopt GrabCut [30], a state-of-the-art interactive segmentation algorithm based on graph cut, to roughly crop out the foreground, and the silhouettes are then extracted from the segmentation results. The approach proposed in Section 3.5 is then used to reconstruct 3D shapes. Fig. 13 illustrates some results on the photos of humans dressed in relatively

Table 5: Quantitative comparisons on human body data given frontal-view and side-view silhouettes. F: use the frontal-view silhouette only; F+S: use both frontal-view and side-view silhouettes. Errors and standard deviations are given in millimeters, and the actual body height of each instance is assumed to be 1700mm.

Data Set		The 1st Candidate			
		GPLVM-I (F)	GPLVM-I (F+S)	PCA (F+S)	NN (F+S)
Dimensionality		8	8	20	
Humans (mixed)	Chest	16.3 ± 12.9	8.8 ± 5.8	**8.0 ± 5.6**	13.6 ± 10.9
	Waist	15.7 ± 10.6	**9.2 ± 5.1**	9.4 ± 4.7	16.0 ± 12.2
Humans (female)	Chest	16.0 ± 10.7	**7.4 ± 5.1**	7.8 ± 4.8	12.4 ± 9.4
	Waist	15.1 ± 9.5	9.4 ± 4.5	**8.6 ± 4.6**	14.2 ± 9.6

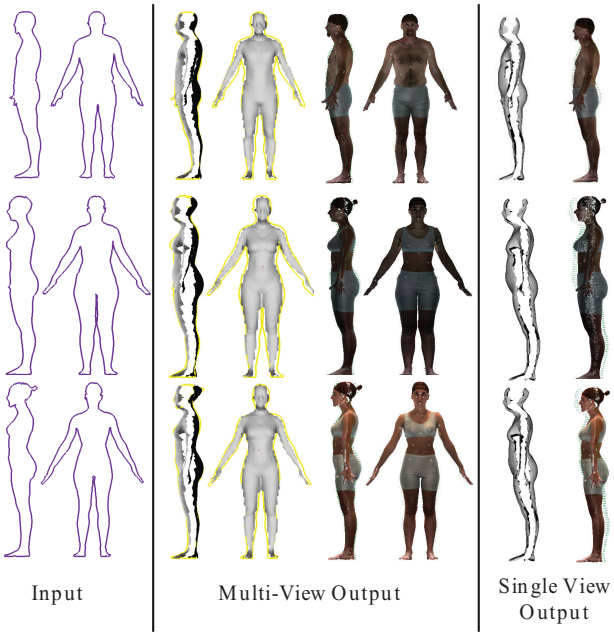


Figure 12: Reconstruction from both frontal-view and side-view silhouettes of the human body. For each pair of given testing input silhouettes (column 1), we show the reconstructed surface in column 2 in contrast with the ground truth. We also compare it with the reconstruction result based on a single frontal-view silhouette (column 3).

tight-fitting clothes. It shows that both single-frontal-view and frontal+side-view settings of ours can give meaningful results. It can be seen that our framework is somewhat robust to imperfect silhouette inputs and outliers.

5. Discussion and Future Work

Experiments show that our approaches work well on the data presented in the previous Section 4. In this section, we discuss some limitations of the approaches presented in this paper.

In the single view reconstruction, the frontal silhouette does not convey enough information for precisely inferring the depth distribution and this usually results in ambiguity in the 3D structure. For most categories of 3D objects, multi-modality is common in the predictive posterior as the mug example in Fig. 8 shows, i.e., the instances with a similar silhouette may have strikingly different depth distributions, corresponding to multiple local maxima of the predictive posterior. In this case, our approach will provide multiple very different candidate solutions. In the example of Fig. 14(a), the first candidate is unsatisfactory. A lady with a relative wider waist is more likely to have a thicker belly in the prediction of our model, but it is not the case of this specific example. This ambiguity can be solved by introducing another side view, as described in Section 4.5, or incorporating more visual information such as texture and color, which we plan to address in the future.

In Section 4.3, it is mentioned that uncertainty measurements indicate that the prediction accuracy varies with the location. We can observe that uncertainty values are usually low at smooth areas such as torso, belly, and thighs. However, at those fine ending parts such as the head, hands and feet of the human body, and discontinuous edges, e.g., the handle of the mug (see Fig. 7 and 9), uncertainty values are high, which indicates poor prediction and explains the artifacts at those areas. An important cause of this phenomenon is that depth maps are usually under-sampled in these areas during the model training stage when a uniform template grid is adopted. Our future work will address this issue, and a part-based representation of shapes is of interest here. Through learning the shape prior of each part, a divide-and-conquer strategy could be applied such that the local shape modeling can be more accurate.

At present, we assume all silhouettes are obtained

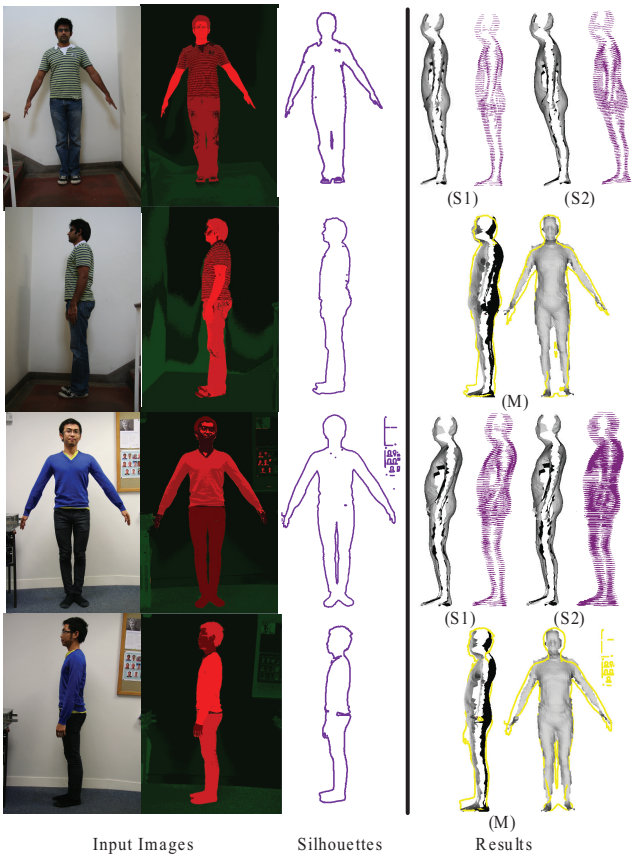
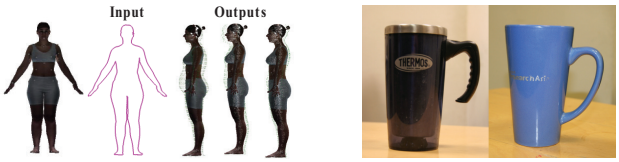


Figure 13: Test on the image inputs. Column 1: image inputs in the frontal and side view; Column 2: segmented foreground and background; Column 3: extracted silhouettes; (S1) and (S2): two highest-posterior candidates of reconstructed body shapes from the single frontal-view silhouette along with their error measurements; (M): reconstructed body shapes from both frontal-view and side-view silhouettes.

in a canonical standing pose for the human body data. We have not taken into account the articulation pose changes of those highly deformable objects such as human body or quadrupeds, and large perspective viewpoint changes of the input. Presumably, a complete shape model in our framework is expected to learn these variations also as a part of latent factors and thus be adapted to input changes. Therefore, another important research issue for us in the near future is to investigate how these variations in the input data influence our reconstruction approach and fit our framework to various types of inputs.

Finally, since we use frontal and dorsal depth maps for 3D shape representation, our current approach is not dealing with internal structures, e.g., the inside of the mug. Currently, we assume that those objects for training the shape model are aligned into a viewpoint with the least occlusion. Also, since our current framework capture continuous variations in the shape space, it is not suitable for modeling sudden topological changes within the category, e.g., a mug with an open handle (see Fig. 14(b)). Future possible solutions to these problems can be voxel-based or level-set-based representations.



(a) (b)

Figure 14: Failure examples. (a) The model generates an unsatisfactory top candidate, although the second and the third ones look reasonable. (b) Mugs may have topological changes around their handles, which cannot be modeled by the current approach.

6. Conclusion

In this paper, we propose a novel framework for learning the shape prior with a Gaussian Process Latent Variable Model and reconstructing dense 3D shapes from 2D single-view/sparse-view silhouettes. Compared with previous methods, our approaches do not depend on any predefined parametrical model and heuristic regularity. A significant advantage of the framework of learning-based reconstruction that we propose is that it can be easily generalized to deal with various categories of 3D objects that may have complex geometrical and topological structures just by simply adjusting the dimension of the latent space in the model. The extension of the current research may include: 1) expanding the current framework to incorporate multiple visual cues to achieve more accurate reconstruction; 2) to further cope with articulation and pose changes of highly deformable input data; 3) investigating shape priors of parts to improve the local shape modeling; 4) conducting more thorough experiments over a wider range of objects.

[1] M. Prasad, A. Zisserman, A. Fitzgibbon, Single view reconstruction of curved surfaces, In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2:1345–1354, 2006.

[2] D. Hoiem, A. Efros, M. Hebert, Automatic photo pop-up, *SIGGRAPH*, 577–584, 2005.

[3] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, *SIGGRAPH*, 187–194, 1999.

[4] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, SCAPE: Shape completion and animation of people, *SIGGRAPH*, 408–416, 2005.

[5] L. Sigal, A. Bălan, M. Black, Combined discriminative and generative articulated pose and non-rigid shape estimation, *Advances in Neural Information Processing System*, 2007.
[6] N. Lawrence, Gaussian process latent variable models for visualisation of high dimensional data, *Advances in Neural Information Processing Systems*, 16:329–336, 2004.
[7] A. Gupta, T. Chen, F. Chen, D. Kimber, L. Davis, Context and observation driven latent variable model for human pose estimation, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
[8] S. Hou, A. Galata, F. Caillette, N. Thacker, P. Bromiley, Real-time body tracking using a gaussian process latent variable model, In *Proc. Int. Conf. on Computer Vision*, 2007.
[9] R. Navaratnam, A. Fitzgibbon, R. Cipolla, Semi-supervised joint manifold learning for multi-valued regression, In *Proc. Int. Conf. on Computer Vision*, 2007.
[10] A. Shon, K. Grochow, A. Hertzmann, R. Rao, Learning shared latent structure for image synthesis and robotic imitation, *Advances in Neural Information Processing Systems*, 16:1233–1240, 2006.
[11] M. Salzmann, R. Urtasun, P. Fua, Local deformation models for monocular 3D shape recovery, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
[12] A. Criminisi, I. Reid, A. Zisserman, Single view metrology, *Int. Journal of Computer Vision*, 40 (2):123–148, 2000.
[13] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, A. Konushin, Fast automatic single-view 3-d reconstruction of urban scenes, In *Proc. European Conf. on Computer Vision*, LNCS 5303:100–103, 2008.
[14] E. Delage, H. Lee, A. Ng, A dynamic bayesian network model for autonomous 3D reconstruction from a single indoor image, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* 2:2418–2428, 2006.
[15] A. Saxena, S. Chung, A. Ng, 3-D depth reconstruction from a single still image, *Int. Journal of Computer Vision*, 76 (1):53–69, 2008.
[16] Y. Chen, J. Liu, X. Tang, A divide-and-conquer approach to 3D object reconstruction from line drawings, In *Proc. Int. Conf. on Computer Vision*, 2007.
[17] J. Liu, L. Cao, Z. Li, X. Tang, Plane-based optimization for 3D object reconstruction from single line drawings, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30 (2) (2008) 315–327.
[18] H. Lipson, M. Shpitalni, Optimization-based reconstruction of a 3D object from a single freehand line drawing, *Computer-Aided Design*, 28 (8):651–663, 1996.
[19] L. Zhang, G. Dugas-Phocion, J. Samson, Single-view modelling of free-form scenes, *The Journal of Visualization and Computer Animation*, 13:225–235, 2002 .
[20] P. Guan, A. Weiss, A. Bălan, M. Black, Estimating human shape and pose from a single image, In *Proc. Int. Conf. on Computer Vision*, 2009.
[21] T. Hassner, R. Basri, Example based 3D reconstruction from single 2D images, *Beyond Patches Workshop at CVPR*, 2006.
[22] F. Han, S. Zhu, Bayesian reconstruction of 3D shapes and scenes from a single image, In *Proc. 1st IEEE Int. Workshop on Higher-Level Knowledge*, 2003.
[23] D. Rother, G. Sapiro, Seeing 3D objects in a single 2D image, In *Proc. Int. Conf. on Computer Vision*, 2009.
[24] L. Torresani, A. Hertzmann, C. Bregier, Nonrigid structure-from-motion: Estimating shape and motion with hierachical priors, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30 (5):878–892, 2008.
[25] M. Moller, A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks*, 6:525–533, 1993.
[26] J. Tenenbaum, Mapping a manifold of perceptual observations, *Advances in Neural Information Processing System*, 10:682–688, 2004.
[27] A. Thayananthan, B. Stenger, P. Torr, R. Cipolla, Shape context and chamfer matching in cluttered scenes, In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1:127–133, 2003.
[28] D. Huttenlocher, J. Noh, W. Rucklidge, Tracking non-rigid objects in complex scenes, In *Proc. Int. Conf. on Computer Vision*, 93–101, 1993.
[29] N. Lawrence, M. Seeger, R. Herbrich, Fast sparse gaussian process methods: The informative vector machine, *Advances in Neural Information Processing Systems*, 15:625–632, 2003.
[30] C. Rother, V. Kolmogorov, A. Blake, "GrabCut" – Interactive foreground extraction using iterated graph cuts, *SIGGRAPH*, 309–314, 2004.

Appendix A. Gradients and Derivatives in Section 3.3 and 3.4

In the model training stage (Section 3.3), generally speaking, there is no closed-form solution to the optimization problem in (13) when general non-linear kernels $\mathbf{K_Y}$ and $\mathbf{K_Z}$ are given and there are likely to be multiple local optima [6]. We thus resort to a gradient-based method to optimize the objective function L.

According to the chain rule, the gradients of L with respect to the latent positions $\mathbf{X}$ and the hyperparameters $\theta_\mathbf{Y}$ and $\theta_\mathbf{Z}$ can be calculated by first taking gradient with respect of the kernels $\mathbf{K_Y}$ and $\mathbf{K_Z}$ and then combining them with $\frac{\partial \mathbf{K_Y}}{\partial \theta_\mathbf{Y}}$, $\frac{\partial \mathbf{K_Z}}{\partial \theta_\mathbf{Z}}$, $\frac{\partial \mathbf{K_Y}}{\partial x_{i,j}}$, and $\frac{\partial \mathbf{K_Z}}{\partial x_{i,j}}$, as shown in (A.1), (A.2), and (A.3).

$$\frac{\partial L}{\partial x_{i,j}} = tr\Big((\frac{\partial L}{\partial \mathbf{K_Y}})^T \frac{\partial \mathbf{K_Y}}{\partial x_{i,j}} + (\frac{\partial L}{\partial \mathbf{K_Z}})^T \frac{\partial \mathbf{K_Z}}{\partial x_{i,j}}\Big) + 2x_{i,j}, \tag{A.1}$$

$$\frac{\partial L}{\partial \theta_{Y,j}} = tr\Big((\frac{\partial L}{\partial \mathbf{K_Y}})^T \frac{\partial \mathbf{K_Y}}{\partial \theta_{Y,j}}\Big), \quad j = 1,2,3,4 \tag{A.2}$$

$$\frac{\partial L}{\partial \theta_{Y,j}} = tr\Big((\frac{\partial L}{\partial \mathbf{K_Z}})^T \frac{\partial \mathbf{K_Z}}{\partial \theta_{Z,j}}\Big), \quad j = 1,2,3,4 \tag{A.3}$$

where the kernel gradient matrices $\frac{\partial L}{\partial \mathbf{K_Y}}$ and $\frac{\partial L}{\partial \mathbf{K_Z}}$ are given as:

$$\frac{\partial L}{\partial \mathbf{K_Y}} = -\frac{1}{2}(m\mathbf{K_Y^{-1}} + \mathbf{K_Y^{-T}}\mathbf{Y}\mathbf{Y}^T\mathbf{K_Y^{-T}}), \tag{A.4}$$

$$\frac{\partial L}{\partial \mathbf{K_Z}} = -\frac{1}{2}(m\mathbf{K_Z^{-1}} + \mathbf{K_Z^{-T}}\mathbf{Z}\mathbf{Z}^\mathbf{T}\mathbf{K_Z^{-T}}). \tag{A.5}$$

When "RBF+linear" kernels are adopted, the other derivatives of kernel elements with respect to the hyperparameters and the latent positions have the following

explicit forms.

$$\frac{\partial K_Y^{(i,j)}}{\partial \theta_{Y,1}} = e^{-\frac{\theta_{Y,2}}{2}(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_i-\mathbf{x}_j)}, \tag{A.6}$$

$$\frac{\partial K_Y^{(i,j)}}{\partial \theta_{Y,2}} = \frac{\theta_{Y,1}}{2}(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_i-\mathbf{x}_j)e^{-\frac{\theta_{Y,2}}{2}(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_i-\mathbf{x}_j)}, \tag{A.7}$$

$$\frac{\partial K_Y^{(i,j)}}{\partial \theta_{Y,3}} = -\theta_{Y,3}^{-2}\delta_{ij}, \tag{A.8}$$

$$\frac{\partial K_Y^{(i,j)}}{\partial \theta_{Y,4}} = \mathbf{x}_i^T\mathbf{x}_j, \tag{A.9}$$

$$\left.\frac{\partial K_Y^{(i,j)}}{\partial \mathbf{x}_i}\right|_{\forall i\neq j} = \frac{\theta_{Y,1}\theta_{Y,2}}{2}(\mathbf{x}_i-\mathbf{x}_j)e^{-\frac{\theta_{Y,2}}{2}(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_i-\mathbf{x}_j)} + 2\theta_{Y,4}\mathbf{x}_j, \tag{A.10}$$

$$\frac{\partial K_Y^{(i,i)}}{\partial \mathbf{x}_i} = 2\theta_{Y,4}\mathbf{x}_j, \tag{A.11}$$

$$\frac{\partial K_Z^{(i,j)}}{\partial \theta_{Z,1}} = e^{-\frac{\theta_{Z,2}}{2}(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_i-\mathbf{x}_j)}, \tag{A.12}$$

$$\frac{\partial K_Z^{(i,j)}}{\partial \theta_{Z,2}} = \frac{\theta_{Z,1}}{2}(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_i-\mathbf{x}_j)e^{-\frac{\theta_{Z,2}}{2}(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_i-\mathbf{x}_j)}, \tag{A.13}$$

$$\frac{\partial K_Z^{(i,j)}}{\partial \theta_{Z,3}} = -\theta_{Z,3}^{-2}\delta_{ij}, \tag{A.14}$$

$$\frac{\partial K_Z^{(i,j)}}{\partial \theta_{Z,4}} = \mathbf{x}_i^T\mathbf{x}_j, \tag{A.15}$$

$$\left.\frac{\partial K_Z^{(i,j)}}{\partial \mathbf{x}_i}\right|_{\forall i\neq j} = \frac{\theta_{Z,1}\theta_{Z,2}}{2}(\mathbf{x}_i-\mathbf{x}_j)e^{-\frac{\theta_{Z,2}}{2}(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_i-\mathbf{x}_j)} + 2\theta_{Z,4}\mathbf{x}_j \tag{A.16}$$

$$\frac{\partial K_Z^{(i,i)}}{\partial \mathbf{x}_i} = 2\theta_{Z,4}\mathbf{x}_j. \tag{A.17}$$

In the prediction stage (Section 3.4), the gradients required to minimize the negative log predictive posterior $H = -\log P(\mathbf{x}|\tilde{\mathbf{y}}, \mathbf{Y}, \mathbf{X}, \theta_{\mathbf{Y}})$ in (15) are given as follows:

$$\frac{\partial H}{\partial \mathbf{x}} = \mathbf{x} + \frac{1}{\hat{\sigma_Y^2}}(\tilde{\mathbf{y}}-\hat{\mu_{\mathbf{Y}}})\frac{\partial \hat{\mu_{\mathbf{Y}}}}{\partial \mathbf{x}} - \left(\frac{1}{2\hat{\sigma_Y^4}}(\tilde{\mathbf{y}}-\hat{\mu_{\mathbf{Y}}})(\tilde{\mathbf{y}}-\hat{\mu_{\mathbf{Y}}})^T + \frac{m}{2\hat{\sigma_Y^2}}\right)\frac{\partial \hat{\sigma_Y^2}}{\partial \mathbf{x}} \tag{A.18}$$

$$\frac{\partial \hat{\mu_{\mathbf{Y}}}}{\partial \mathbf{x}} = \mathbf{Y}^T{\mathbf{K_Y}}^{-1}\frac{\partial \mathbf{k_Y}}{\partial \mathbf{x}} \tag{A.19}$$

$$\frac{\partial \hat{\sigma_Y^2}}{\partial \mathbf{x}} = 2\theta_4\mathbf{x} - {\mathbf{k_Y}}^T{\mathbf{K_Y}}^{-1}\frac{\partial \mathbf{k_Y}}{\partial \mathbf{x}} \tag{A.20}$$

Appendix B. Details for Deriving (27) in Section 3.5

We denote $\mathbf{y}$ and $\mathbf{z}$ as m dimensional PCA feature vectors of 2D grid positions and depth maps, respectively. Let $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}$ are $2N_g \times m$ matrices be PCA eigen-vectors of 2D grid positions and depth maps, respectively, where N_g is the grid density of the 2D template. Here, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{B}_1$, and $\mathbf{B}_2$ are all $N_g \times m$ sub-matrices of $\mathbf{A}$ and $\mathbf{B}$, which control x-coordinates, y-coordinates, frontal depth values, and dorsal depth values of all the grid points, respectively. Then, according to (6) and (7), both 2D image positions $\mathbf{G}_y$ and depth maps $\mathbf{G}_z$ of all the grid points are $2N_g$-D column vectors, which can be written as $\mathbf{G}_y = \begin{bmatrix} \mathbf{G}_y^1 \\ \mathbf{G}_y^2 \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{y,0}^1 + \mathbf{A}_1\mathbf{y} \\ \mathbf{G}_{y,0}^2 + \mathbf{A}_2\mathbf{y} \end{bmatrix}$ and $\mathbf{G}_z = \begin{bmatrix} \mathbf{G}_z^1 \\ \mathbf{G}_z^2 \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{z,0}^1 + \mathbf{B}_1\mathbf{z} \\ \mathbf{G}_{z,0}^2 + \mathbf{B}_2\mathbf{z} \end{bmatrix}$, respectively, where $\mathbf{G}_{y,1}$, $\mathbf{G}_{y,2}$, $\mathbf{G}_{z,1}$, $\mathbf{G}_{z,2}$ are the corresponding N_g-D sub-vectors of $\mathbf{G}_y$ and $\mathbf{G}_z$.

Note that $\mathbf{V} = [\mathbf{v}_i]_{i=1}^{2N_g}$ is a $6N_g$-D column vector concatenated by all the $2N_g$ 3D sampling points of the reconstructed object. For the simplicity of notation, we let $\mathbf{e_j^i}$ be the j-D basis vector with the i-th element 1 and all other elements 0, and $\mathbf{I_j^i}$ be the $j \times j$ basis matrix with the i-th diagonal element 1 and all other elements 0, and $\otimes$ denotes the Kronecker product. $\mathbf{V}$ can thus be written into the following matrix formulation.

$$\begin{aligned}\mathbf{V} &= \mathbf{G}_y^1 \otimes (\mathbf{e_6^1} + \mathbf{e_6^4}) + \mathbf{G}_y^2 \otimes (\mathbf{e_6^2} + \mathbf{e_6^5}) \\ &\quad + \mathbf{G}_z^1 \otimes \mathbf{e_6^3} + \mathbf{G}_z^2 \otimes \mathbf{e_6^6} \\ &= (\mathbf{G}_{y,0}^1 + \mathbf{A}_1\mathbf{y}) \otimes (\mathbf{e_6^1} + \mathbf{e_6^4}) \\ &\quad + (\mathbf{G}_{y,0}^2 + \mathbf{A}_2\mathbf{y}) \otimes (\mathbf{e_6^2} + \mathbf{e_6^5}) \\ &\quad + (\mathbf{G}_{z,0}^1 + \mathbf{B}_1\mathbf{z}) \otimes \mathbf{e_6^3} + (\mathbf{G}_{z,0}^2 + \mathbf{B}_2\mathbf{z}) \otimes \mathbf{e_6^6} \\ &= \tilde{\mathbf{G}}_\mathbf{0} + \tilde{\mathbf{A}}\mathbf{\Lambda}. \end{aligned} \tag{B.1}$$

where

$$\begin{aligned}\tilde{\mathbf{G}}_\mathbf{0} &= \mathbf{G}_{y,0}^1 \otimes (\mathbf{e_6^1} + \mathbf{e_6^4}) + \mathbf{G}_{y,0}^2 \otimes (\mathbf{e_6^2} + \mathbf{e_6^5}) \\ &\quad + \mathbf{G}_{z,0}^1 \otimes \mathbf{e_6^3} + \mathbf{G}_{z,0}^2 \otimes \mathbf{e_6^6} \end{aligned} \tag{B.2}$$

$$\begin{aligned}\tilde{\mathbf{A}} &= \mathbf{A}_1 \otimes (\mathbf{I_6^1} + \mathbf{I_6^4}) + \mathbf{A}_2 \otimes (\mathbf{I_6^2} + \mathbf{I_6^5}) \\ &\quad + \mathbf{B}_1 \otimes \mathbf{I_6^3} + \mathbf{B}_2 \otimes \mathbf{I_6^6} \end{aligned} \tag{B.3}$$

$$\mathbf{\Lambda} = \mathbf{y} \otimes (\mathbf{e_6^1} + \mathbf{e_6^2} + \mathbf{e_6^4} + \mathbf{e_6^5}) + \mathbf{z} \otimes (\mathbf{e_6^3} + \mathbf{e_6^6}). \tag{B.4}$$

From (B.1), it is obvious that $\mathbf{V}$ is a linear combination of $\mathbf{y}$ and $\mathbf{z}$, which both have Gaussian forms when the q-D latent coordinate $\mathbf{x}$ of the 3D reconstruction is

given, i.e.,

$$P(\mathbf{y}|\mathbf{x},\mathcal{M}) = \mathcal{N}(\mathbf{y};\mu_{\mathbf{y}}(\mathbf{x}),\sigma_y^2(\mathbf{x})\mathbf{I}_{\mathbf{m}\times\mathbf{m}}), \tag{B.5}$$

$$P(\mathbf{z}|\mathbf{x},\mathcal{M}) = \mathcal{N}(\mathbf{z};\mu_{\mathbf{z}}(\mathbf{x}),\sigma_z^2(\mathbf{x})\mathbf{I}_{\mathbf{m}\times\mathbf{m}}), \tag{B.6}$$

where the formulations of $\mu_{\mathbf{y}}$, σ_y^2, $\mu_{\mathbf{z}}$, and σ_z^2 can be found in (16), (17), (20), and (21), respectively. We can finally write $P(\mathbf{V}|\mathbf{x},\mathcal{M})$ by applying the property of Gaussian distributions.

$$P(\mathbf{V}|\mathbf{x},\mathcal{M}) = \mathcal{N}(\mathbf{V};\mu_{\mathbf{V}}(\mathbf{x}),\mathbf{\Sigma}_{\mathbf{V}}(\mathbf{x})), \tag{B.7}$$

where $\mu_V = \tilde{\mathbf{G}}_\mathbf{0} + \tilde{\mathbf{A}}\mathbf{\Lambda}_\mu$, $\mathbf{\Sigma}_V = \tilde{\mathbf{A}}\mathbf{\Lambda}_\mathbf{\Sigma}\tilde{\mathbf{A}}^T$, and here

$$\mathbf{\Lambda}_\mu = \mu_{\mathbf{y}} \otimes (\mathbf{e}_\mathbf{6}^\mathbf{1} + \mathbf{e}_\mathbf{6}^\mathbf{2} + \mathbf{e}_\mathbf{6}^\mathbf{4} + \mathbf{e}_\mathbf{6}^\mathbf{5}) + \mu_{\mathbf{z}} \otimes (\mathbf{e}_\mathbf{6}^\mathbf{3} + \mathbf{e}_\mathbf{6}^\mathbf{6}), \tag{B.8}$$

$$\begin{aligned}\mathbf{\Lambda}_\mathbf{\Sigma} = {} & \sigma_y^2\mathbf{I}_{\mathbf{m}\times\mathbf{m}} \otimes ((\mathbf{e}_\mathbf{6}^\mathbf{1} + \mathbf{e}_\mathbf{6}^\mathbf{2} + \mathbf{e}_\mathbf{6}^\mathbf{4} + \mathbf{e}_\mathbf{6}^\mathbf{5})(\mathbf{e}_\mathbf{6}^\mathbf{1} + \mathbf{e}_\mathbf{6}^\mathbf{2} + \mathbf{e}_\mathbf{6}^\mathbf{4} + \mathbf{e}_\mathbf{6}^\mathbf{5})^T) \\ & + \sigma_z^2\mathbf{I}_{\mathbf{m}\times\mathbf{m}} \otimes ((\mathbf{e}_\mathbf{6}^\mathbf{3} + \mathbf{e}_\mathbf{6}^\mathbf{6})(\mathbf{e}_\mathbf{6}^\mathbf{3} + \mathbf{e}_\mathbf{6}^\mathbf{6})^T).\end{aligned} \tag{B.9}$$

On the other hand, since projecting the 3D shape $\mathbf{V}$ into the k-th view is a linear process according to (25), the silhouette $\mathbf{W_k}$ (a $2N_k^b$-D column vector) can also be written as a linear combination of $\mathbf{y}$ and $\mathbf{z}$.

$$\mathbf{W_k} = \tilde{\mathbf{P}}_\mathbf{k}^\mathbf{T}\mathbf{V} + \tilde{\mathbf{t}}_\mathbf{k} + \mathbf{n_k} = \tilde{\mathbf{P}}_\mathbf{k}^\mathbf{T}(\tilde{\mathbf{G}}_\mathbf{0} + \tilde{\mathbf{A}}\mathbf{\Lambda}) + \tilde{\mathbf{t}}_\mathbf{k} + \mathbf{n_k}, \tag{B.10}$$

where $\tilde{\mathbf{P}}_\mathbf{k}^\mathbf{T}$ and $\tilde{\mathbf{t}}_\mathbf{k}$ are defined in Section 3.5, and $\mathbf{n_k}$ is the Gaussian noise subjected to $N(\mathbf{n_k};\mathbf{0},\sigma_w^2\mathbf{I}_{\mathbf{2N_k^b}\times\mathbf{2N_k^b}})$. Hence, the likelihood $P(\mathbf{W_k}|\mathbf{x},\mathcal{M},\gamma_\mathbf{k})$ is also a Gaussian distribution:

$$P(\mathbf{W_k}|\mathbf{x},\mathcal{M},\gamma_\mathbf{k}) = \mathcal{N}(\mathbf{W_k};\mu_{\mathbf{W_k}}(\mathbf{x},\gamma_\mathbf{k}),\mathbf{\Sigma}_{\mathbf{W_k}}(\mathbf{x},\gamma_\mathbf{k})), \tag{B.11}$$

where $\mu_{\mathbf{W_k}} = \tilde{\mathbf{P}}_\mathbf{k}^\mathbf{T}(\tilde{\mathbf{G}}_\mathbf{0} + \tilde{\mathbf{A}}\mathbf{\Lambda}_\mu) + \tilde{\mathbf{t}}_\mathbf{k}$, and $\mathbf{\Sigma}_{\mathbf{W_k}} = \tilde{\mathbf{P}}_\mathbf{k}^\mathbf{T}\tilde{\mathbf{A}}\mathbf{\Lambda}_\mathbf{\Sigma}\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{P}}_\mathbf{k} + \sigma_w^2\mathbf{I}_{\mathbf{2N_k^b}\times\mathbf{2N_k^b}}$.

Appendix C. Efficient Computation of the Determinant in (31)

The determinant $\det(\mathbf{I} + \frac{1}{N_k^b\sigma_s^2}\mathbf{\Sigma}_{\mathbf{W_k}})$ in (31) can be factorized into the following form:

$$\det(\mathbf{I} + \frac{1}{N_k^b\sigma_s^2}\mathbf{\Sigma}_{\mathbf{W_k}}) = \prod_{i=1}^{2N_k^b}\Big(\frac{\lambda_{D,i}+\sigma_w^2}{N_k^b\sigma_s^2} + 1\Big), \tag{C.1}$$

where $\lambda_{D,i}$ $(i = 1, 2, \cdots, 2N_k^b)$ denote the eigen-values of the $2N_k^b \times 2N_k^b$ matrix $\mathbf{D} = \tilde{\mathbf{P}}_\mathbf{k}^\mathbf{T}\tilde{\mathbf{A}}\mathbf{\Lambda}_\mathbf{\Sigma}\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{P}}_\mathbf{k}$.

Directly solving the eigen-decomposition of $\mathbf{D}$ can be expensive. Instead, we work on its $6m \times 6m$ dual matrix $\tilde{\mathbf{D}} = \mathbf{\Lambda}_\mathbf{\Sigma}^\mathbf{1/2}\tilde{\mathbf{A}}^\mathbf{T}\tilde{\mathbf{P}}_\mathbf{k}\tilde{\mathbf{P}}_\mathbf{k}^\mathbf{T}\tilde{\mathbf{A}}\mathbf{\Lambda}_\mathbf{\Sigma}^\mathbf{1/2}$, where $\mathbf{\Lambda}_\mathbf{\Sigma}^\mathbf{1/2}$ represents the Cholesky decomposition of the diagonal matrix $\mathbf{\Lambda}_\mathbf{\Sigma}$. According to the linear algebra theory, the eigen-values $\tilde{\lambda}_{D,i}$ $(i = 1, 2, \cdots, 6m)$ of $\tilde{\mathbf{D}}_\mathbf{k}$ satisfy that $\lambda_{D,i} = \begin{cases} \tilde{\lambda}_{D,i}, & i = 1, 2, \cdots, 6m; \\ 0, & i = 6m+1, 6m+2, \cdots, 2N_k^b. \end{cases}$. It follows that

$$\begin{aligned}&\det(\mathbf{I} + \frac{1}{N_k^b\sigma_s^2}\mathbf{\Sigma}_{\mathbf{W_k}}) \\ =&\Big(\frac{\sigma_w^2}{N_k^b\sigma_s^2} + 1\Big)^{2N_k^b-6m} \cdot \prod_{i=1}^{6m}\Big(\frac{\tilde{\lambda}_{D,i}+\sigma_w^2}{N_k^b\sigma_s^2} + 1\Big).\end{aligned} \tag{C.2}$$

However, $\tilde{\lambda}_{D,i}$ $(i = 1, 2, \cdots, 6m)$ are much faster to compute since $6m \ll 2N_k^b$. The overall complexity of computing the determinant is hence reduced from $O({N_k^b}^3)$ to $O(m^3)$.