Shared Shape Spaces - Draft Version

Victor Adrian Prisacariu
University of Oxford
victor@robots.ox.ac.uk

Ian Reid
University of Oxford
ian@robots.ox.ac.uk

Abstract

We propose a method for simultaneous shape-constrained segmentation and parameter recovery. The parameters can describe anything from 3D shape to 3D pose and we place no restriction on the topology of the shapes, i.e. they can have holes or be made of multiple parts. We use Shared Gaussian Process Latent Variable Models to learn multimodal shape-parameter spaces. These allow non-linear embeddings of the high-dimensional shape and parameter spaces in low dimensional spaces in a fully probabilistic manner. We propose a method for exploring the multimodality in the joint space in an efficient manner, by learning a mapping from the latent space to a space that encodes the similarity between shapes. We further extend the SGP-LVM to a model that makes use of a hierarchy of embeddings and show that this yields faster convergence and greater accuracy over the standard non-hierarchical embedding. Shapes are represented implicitly using level sets, and inference is made tractable by compressing the level set embedding functions with discrete cosine transforms. We show state of the art results in various fields, ranging from pose recovery to gaze tracking and to monocular 3D reconstruction.

1. Introduction

Many computer vision tasks involve learning a mapping between a shape and a parameter space. These parameters vary from the pose of a person generating a set of silhouettes [5, 9], to the position on the screen the person is looking at (as a function of the sclera of the eyes), or to the 3D model of an object (as a function of its pose or shape) [10].

Learning such a mapping usually involves finding the shape, extracting some features from it and computing the value for the unknown parameter(s), using the features. The final result depends on a good contour extraction step and on the ability of the descriptor to capture enough of the variance from the shape space, neither of which can be guaranteed in practice. Our aim is to perform *simultaneous* shape-constrained segmentation and parameter recovery. Previous work has addressed the two in isolation [1, 5], and usually in a discriminative manner, by learning a mapping from shape to features. This approach does not however account for multimodality i.e. multiple sets of parameters can correspond to the same shape. For example multiple poses can generate the same contour, or multiple 3D models can project to the same silhouette. One solution is to use a generative approach i.e. rather than learn the conditional distribution between the descriptor and the parameters, learn their joint distribution. As it is difficult to learn the full high dimensional mapping, it is usually assumed that the features, parameters and mapping lie on a lower dimensional manifold. A probabilistic and nonlinear method of learning such a manifold is Shared Gaussian Process Latent Variable Models [5, 9](SGP-LVM).

In this work we show that, through use of *shared* mappings learned using SGP-LVM, we can simultaneously find a segmentation that is consistent with a pre-learned set of shapes, and recover the generating parameters. Our work has similarity to [11], which showed how explicit contours can be generated from GP-LVM spaces, by representing them with elliptic Fourier descriptors. They also proposed a method for using these contours as shape priors for level set based segmentation i.e. find the latent space point which generates the contour that minimises a level set-based, image-driven, energy function. A major issue with this approach is the use of the explicit representation of the contour, as it cannot easily model holes. Also sudden changes in the topology of the shape often lead to discontinuities in the latent space (which make convergence more difficult). Implicit representations of the contours have been used throughout the segmentation literature as shape priors in segmentation [3, 15], but learning them directly using a GP-LVM is *very* time and memory consuming.

In this paper we use SGP-LVM to learn generative joint shape–parameter spaces. We represent shapes implicitly, using discrete cosine transforms of level set embedding functions. Inference has two steps: (i) minimise a level set-based, image-

driven, energy function w.r.t. a latent point and (ii) project the result to the parameter space. The shared space also enables us to learn the multimodality effectively. Finally, rather than using a single low dimensional shared manifold, as most other work does, we use a hierarchy of such manifolds, with different dimensionalities.

Our method has the following advantages over previous work: (i) we can learn generative multimodal shape-parameter spaces; (ii) we can often search for the modes of the learned spaces in a more direct manner than using the standard exhaustive strategy; (iii) inference requires only two steps and does not need any prior segmentation or any intermediary feature descriptor; (iv) we can model all types of shapes (with and without holes, with sudden topology changes), while still keeping learning and inference tractable; (v) we can use a higher number of dimensions for the learned manifold, with a smaller risk of running into local minima.

A standard application of our method is pose recovery and we demonstrate state of the art results. We also consider a variety of other applications showing that, for instance, we can recover eye gaze as a function of the segmentation of the sclera, or 3D car shape as a function of the 2D segmentation, even in the presence of occlusions and missing parts.

The remainder of this paper is structured as follows: we present our learning step in Section 2 and our inference step in Section 3. We show various test results in Section 4 and conclude in Section 5.

2. Learning

Our task is to learn a mapping from a set of n shapes $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ to a set of n parameters $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$, where these parameters could represent a variety of things, from pose joint angles to a 3D model.

We cast this problem in a generative framework, using *Gaussian Processes Latent Variable Models* (GP-LVM) [7], a probabilistic non-linear dimensionality reduction technique.

Given a $[\mathbf{Y}, \mathbf{Z}]$ data set, composed of $(\mathbf{y}_i, \mathbf{z}_i)$ pairs, we use GP-LVM to find the shared set of latent variables $\mathbf{X}_{YZ} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ that captures as much of the variance in the $[\mathbf{Y}, \mathbf{Z}]$ data set as possible. We also learn separate Gaussian Processes (GPs) [12], mapping from the shared to each observation space. This is done by maximising:

$$P(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \theta_{\mathbf{Y}}, \theta_{\mathbf{Z}}) = P(\mathbf{Y}|\mathbf{X}, \theta_{\mathbf{Y}})P(\mathbf{Z}|\mathbf{X}, \theta_{\mathbf{Z}}) \tag{1}$$

with respect to $\mathbf{X}$ (the low dimensional latent variables), $\theta_{\mathbf{Y}}$ and $\theta_{\mathbf{Z}}$ (the hyperparameters of the GPs mapping to the observation spaces). This method is known as shared GP-LVM [4], and has the graphical model of Figure 1.

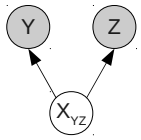

Figure 1. Shared GP-LVM graphical model. $\mathbf{Y}$ and $\mathbf{Z}$ are the observation spaces (the shape and the parameter spaces in our case) and $\mathbf{X}_{YZ}$ is the lower dimensional joint latent space.

In order to force the one-to-one mapping between latent space points and observation space points, we use the same method as [8, 4], and represent the latent variables as smooth mappings (called back-constraints) from the observation spaces, and optimise w.r.t. the parameters of the back-constraints, rather than w.r.t. the latent points directly.

This formulation addresses our two main objectives. As shown in [11], given an image of an object and a GP-LVM latent space of object silhouettes, it is possible to find a constrained segmentation of that object by minimising an image-based error directly in that latent space (so w.r.t. the latent space position). By learning a shared space of both shape and parameters (rather than just a private space for shapes) we can simply back-project the latent space position we found in the segmentation to the space of parameters, therefore finding the set of parameters corresponding to the segmented shape. This formulation also allows for multimodal mappings, as a latent space point space can generate different shapes (values of $\mathbf{y}_i$) for the same set of parameters (values of $\mathbf{z}_i$).

There are however several issues which need to be addressed in order for us to be able to perform useful and timely inference. First, while this model accounts for multimodality, searching for the modes is very inefficient. Second, the choice for the number of dimensions of the latent space is often empirical and dependant on the dataset. Finally, the shape representation used in [11] does not easily accommodate holes or sudden changes in topology. We address these issues in this section.

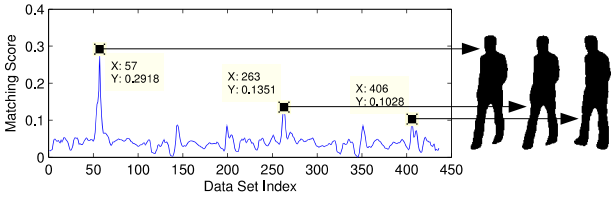

Figure 2. Example similarity function

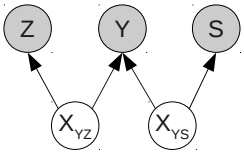

Figure 3. Graphical model for our modes discovery stage. $\mathbf{Y}$ is the shape space, $\mathbf{S}$ the similarity space, $\mathbf{X}_{YS}$ the shared space between shape and shape similarity and $\mathbf{X}_{YZ}$ the shaped space between shapes and sets of parameters.

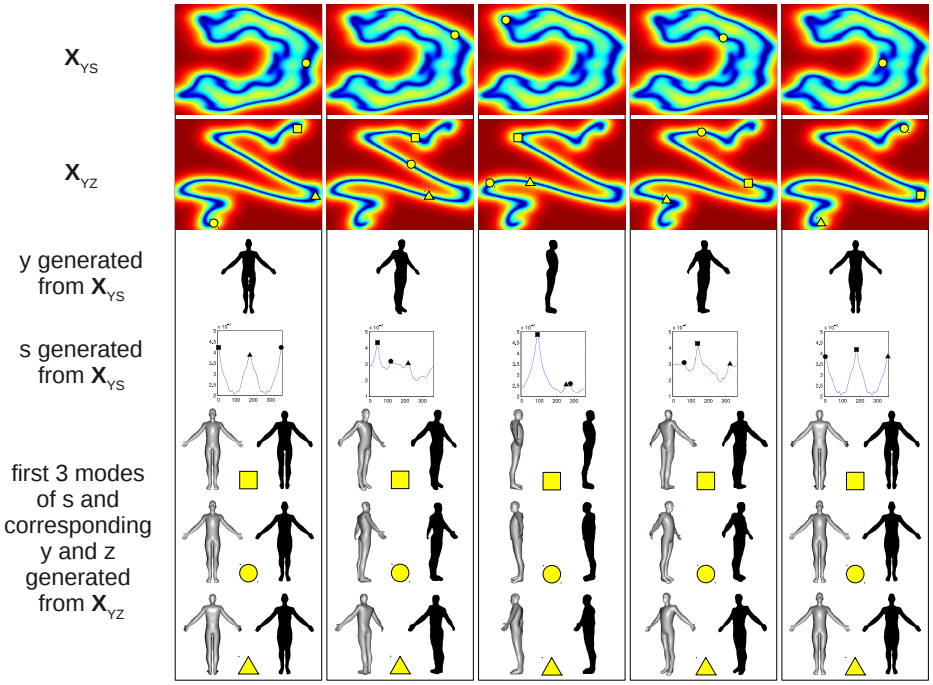

Figure 4. Example of our method discovering all the modes in the silhouette space of a person rotating 360 degrees on a single axis. The point marked in $\mathbf{X}_{YS}$ generates a shape $\mathbf{y}$ and a function $s_{\mathbf{y}}$ (3rd and 4th rows). The top 3 peaks of $s_{\mathbf{y}}$ indicate the top 3 $\mathbf{x}_{YZ}$ points that are likely to generate similar shapes (2nd row). The last 3 rows show the back-projections of the peaks to both shape and pose spaces.

2.1. Manifold Modes Discovery

SGP-LVM can generate similar shapes for different latent points. Our goal is to find all these points. The standard approach to this task is to perform optimisation in the latent space, by running several searches (minimisations of an energy function), initialised from multiple random points in the latent space. While obviously inefficient, this method is the only one to work on all types of data sets.

In many cases however, even though there are multiple modes in the mapping, there is a common source for the multi-modality. For example, even though multiple 3D objects can project to similar 2D shapes, the ones that do often belong to the same class of 3D object. If a symmetric 3D object is rotated around an axis it produces similar 2D projections because of that symmetry. In a person's walk, it is the symmetry of the action that leads to similar silhouettes.

In this work, to model the above-mentioned behaviour, we aim to learn a function that, for any given shape, tells us what are the other similar shapes in the training data set. Such a function is shown in Figure 2. For each shape in the data set, we

compute a probability distribution describing the similarity between that shape and the others in the data set, using a standard similarity measure (in our case area overlap). The number of sample points for this pdf is equal to the number of points in the data set. For larger data sets approximations of this distribution could be used. For two shapes $\mathbf{y}_i$ and $\mathbf{y}_j$ in $\mathbf{Y}$ we can write this formally as:

$$s_{\mathbf{y}_i}(\mathbf{y}_j) = \frac{similarity(\mathbf{y}_i, \mathbf{y}_j)}{\sum_{k \in \mathbf{Y}} similarity(\mathbf{y}_i, \mathbf{y}_k)} \tag{2}$$

We then learn a shared GP-LVM space between each shape and its corresponding $s_{\mathbf{y}_i}$. Figure 3 shows the graphical model for this part of the model. The joint space $\mathbf{X}_{YS}$ generates both the shape space $\mathbf{Y}$ and the similarity space $\mathbf{S}$. Thus, for any given latent point $\mathbf{x}_{ys}$ we can generate (i) a shape $\mathbf{y}_i$, and (ii) a function $s_{\mathbf{y}_i}$ over all shapes, whose modes are shapes that are similar to $\mathbf{y}_i$.

At inference time, to find the most similar shapes from the training data set, for a shape $\mathbf{y}$, we project the latent space point generating that shape to the space of similarity functions and find the peaks in that function, i.e. the set of indices i:

$$\{i \mid s_y(y_i) \text{ is locally maximum}\} \tag{3}$$

Figure 4 shows the results obtained by our method when used to find the modes in the space of shapes produced by rotating a person 360 degrees on a single axis. The point marked with the yellow circle in the $\mathbf{X}_{YS}$ space generates both a shape $\mathbf{y}$ (3rd row) and a function $s_{\mathbf{y}}$ (4th row). The top 3 peaks of this function indicate the top 3 points in the $\mathbf{X}_{YZ}$ latent space that are likely to generate similar shapes (2nd row). The results of back-projecting these peaks to both the shape space and the pose space are shown in the last 3 rows.

2.2. Hierarchy of Manifolds

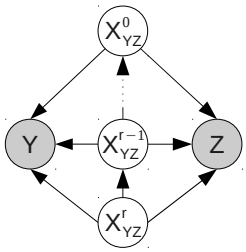

Figure 5. Graphical model for our hierarchy. $\mathbf{Y}$ is the shape space, $\mathbf{Z}$ the parameter space and $[\mathbf{X}^0_{YZ}, \ldots, \mathbf{X}^r_{YZ}]$ the hierarchical shared shape – parameter spaces.

It is an open problem how best to choose the dimensionality of the latent manifold. A smaller number of dimensions leads to a denser latent space, with an easier inference, but this may not capture the variability in the data adequately and consequently have less generative accuracy. A higher number of dimensions has greater generative accuracy, but at the cost of a sparser latent space with more difficult inference. Almost all previous work (to our knowledge) chooses a unique dimension based on the amount of variance needed to be captured or on empirical observations about the data set. For example [9] uses 2 dimensions, because the motion to be learned is known to be circular. In most cases however such a deduction is impossible. Furthermore, often, some points in the latent space will require a higher number of dimensions (for reconstruction) than others. This makes it difficult to choose a single, minimum dimensionality, that is optimal for all points in the space.

Our novel approach to this dilemma is to learn a hierarchy of dimensions, as shown in Figure 5. Each $\mathbf{X}^r_{YZ}$ space is a lower dimensional embedding of the $\mathbf{X}^{r-1}_{YZ}$ space. $\mathbf{X}^0_{YZ}$ is the highest dimensional low dimensional embedding for the shared shape – parameter space manifold. We also learn GP mappings from each step of the hierarchy to each observation space (to make inference easier). In this work $r = 3$: we first learn a mapping between the observation data and a 10D space, then between the 10D space and a 4D space and finally between the 4D space and a 2D space.

2.3. Shape Representation

We need a differentiable and invertible shape descriptor, which can represent both holes and sudden changes in topology, while still keeping learning and inference steps tractable. Explicit shape representations (like in [11]) cannot model the appearance and disappearance of holes and can lead to discontinuities in the latent space (which make inference more difficult). A solution to these problems, that has been used extensively throughout the segmentation literature [15, 3], is to represent shapes implicitly, with level set embedding functions (i.e. signed distance functions). Due to their high dimensionality,

building latent spaces directly from the signed distance functions can easily become intractable and is highly prone to local minima.

We use 2D discrete cosine transforms (DCTs) to compress the embedding functions, thereby reducing the dimensionality considerably. For example, for a 640x480 image, the full signed distance function requires 307200 dimensions, while a 35 harmonic 2D DCT requires only 1225 dimensions. Furthermore, since the high frequency components of the signed distance function are used only to encode the "tip" of the function, compression is practically lossless in the band around the contour where we evaluate the energy function.

The 2D DCT is a special case of the discrete Fourier transform, where the input data contains of only real numbers from an even function. It represents an image as a series of cosines, each with its own frequency and amplitude. For forward and reverse DCT formulas see [6].

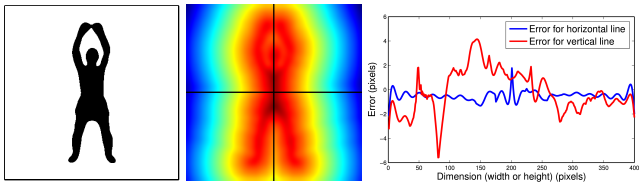

Figure 6. Distance transform accuracy example. The contour is the zero level of the middle distance transform, which is generated from a 2D latent space. The chart shows vertical and horizontal cross sections in the per pixel error between this distance transform and the ground truth one, for the same contour.

Figure 6 shows an example shape and distance transform generated by our method from a 2D latent space of the star-jump motion of [11]. We also computed the actual (ground truth) distance transform of the contour. The chart shows vertical and horizontal cross sections in the per pixel difference between the two distance transforms. The maximum difference is smaller than 6 pixels. The average pixel error, on all shapes in all our datasets, was only *1.41 pixels*.

3. Inference

For any image I, the complete inference proceeds as follows: we first search for the latent space point in $\mathbf{X}_{YS}$ that generates the shape $\mathbf{y}$ that best segments the image. Next we project that point to the similarity space, and obtain a similarity function s, as shown in Subsection 2.1. We select the top n_p peaks in the similarity function, $[i_1, \ldots, i_{n_p}]$, as per Equation 3. n_p is a tunable parameter, which for our examples we have set to 3. While better methods for sampling s exist, this is not the focus of the present work. Each index i corresponds to a latent variable $\mathbf{x}_i$ from the $\mathbf{X}^r_{YZ}$ space and we use $[\mathbf{x}_i, \ldots, \mathbf{x}_{i_{n_p}}]$ as initialisations for n_p other searches, this time in the $\mathbf{X}^r_{YZ}$ latent space, again aiming for an optimal (shape-constrained) segmentation of I. The n_p results of these searches will most likely project to very similar shapes but different values for the parameter space $\mathbf{Z}$. They also project to points in the higher dimensional $\mathbf{X}^{r-1}_{YZ}$ latent space. These projections serve as initialisations for another n_p searches, this time in the $\mathbf{X}^{r-1}_{YZ}$ latent space, again looking to segment the image I. We repeat this operation for each latent space in the hierarchy. The final result is a set of n_p possible segmentations for the image, each projecting to a separate set of parameters, generated from the highest dimensional latent space $\mathbf{X}^0_{YZ}$.

The core of our inference is the minimisation of a level set based, image-driven, energy function, with respect to a latent space point, which finds the low dimensional point that generates a shape that best segments an image. Similar to [11] we define a monotonic and continuously differentiable energy function $f(\Phi, I)$ as a measure of how well the level set embedding function Φ segments I.

We differentiate $f(\Phi, I)$ with respect to the latent variables $\mathbf{x}_i$, $i = [1, \ldots, q]$, using the chain rule:

$$\frac{\partial f}{\partial \mathbf{x}_i} = \frac{\partial f}{\partial \Phi} \frac{\partial \Phi}{\partial \mu} \frac{\partial \mu}{\partial \mathbf{x}_i} \tag{4}$$

We use Gaussian Processes, so each $\mathbf{x}_i$ is actually a normal distribution with a mean $\mu_i = K_* K^{-1} \mathbf{Y}$ and a variance $\upsilon = \sigma^2 = \kappa(\mathbf{x}_i, \mathbf{x}_i) - K_* K^{-1} K_*^T$, with K_* being the covariance between $\mathbf{x}_i$ and all the learned latent points.

$\frac{\partial \mu}{\partial \mathbf{x}_i}$ can be computed as:

$$\frac{\partial \mu}{\partial \mathbf{x}_i} = \frac{\partial \kappa(\mathbf{x}_i, \mathbf{X})}{\partial \mathbf{x}_i} K^{-1} \mathbf{Y} \tag{5}$$

$\frac{\partial \kappa(\mathbf{x}_i, \mathbf{X})}{\partial \mathbf{x}_i}$ follow trivially, since κ is essentially an RBF.

It can be shown that the derivative of the inverse 2D DCT of a matrix is the inverse 2D DCT of the derivative of that matrix. Since Φ is the inverse 2D DCT of μ, it follows that $\frac{\partial \Phi}{\partial \mu}$ is just the inverse 2D DCT of $\frac{\partial \mu}{\partial \mathbf{x}_i}$. This observation allows us to compute $\frac{\partial \Phi}{\partial \mu}$ using the fast inverse Fourier algorithm, with a complexity of O(N log N).

As $f(\Phi, I)$ we use the Bibby-Reid energy function [2] so $f(\Phi, I) = -\log(P(\Phi|\Omega, \mathbf{x}_i))$, where:

$$P(\Phi|\Omega, \mathbf{x_i}) = \prod_{x \in \Omega} \Big\{ H_e\Big(\Phi(x)\Big) P_f + \Big(1 - H_e\big(\Phi(x)\big)\Big) P_b \Big\} P(\Phi|\mathbf{x}_i) \tag{6}$$

Here Φ is the level set embedding function segmenting image I, Ω is the image domain, H_e is the smooth Heaviside function, x is the pixel in the image, and:

$$P_f = \frac{P(y|M_f)}{\eta_f P(y|M_f) + \eta_b P(y|M_b)} \tag{7}$$

with η_f and η_b the number of foreground and background pixels, y the colour of the x and $P(y|M)$ the likelihood of y given that colour model. We use RGB images and our models M_f and M_b are histograms with 32 bins per channel.

We added a term containing the GP-generated variance υ (a function of the latent space position), which contains information regarding the validity of generated shapes: low variance points are more likely to be valid shapes than high variance points: $P(\Phi|\mathbf{x}_i) = \frac{1}{\sigma\sqrt{2\pi}}$. This term might not always lead to the best image fitting shape, but rather to the one with the highest latent space probability. To account for this, after minimising the energy function with this term, we run another 10% of the total number of iterations without it.

Finally, for this $f(\Phi, I)$, $\frac{\partial f}{\partial \Phi}$ is:

$$\frac{\partial f}{\partial \Phi} = -\sum_{x \in \Omega} \frac{\delta(\Phi)(P_f - P_b)}{H_e(\Phi)P_f + \big(1 - H_e(\Phi)\big)P_b} \frac{\partial \Phi}{\partial \mu} + \frac{1}{2\upsilon} \frac{\partial \upsilon}{\partial \mathbf{x}_i} \tag{8}$$

with $\frac{\partial \upsilon}{\partial \mathbf{x}_i}$ being computed as:

$$\frac{\partial \upsilon}{\partial \mathbf{x}_i} = \frac{\partial \kappa(\mathbf{x}_i, \mathbf{x}_i)}{\partial \mathbf{x}_i} - 2 \frac{\partial \kappa(\mathbf{x}_i, \mathbf{X})}{\partial \mathbf{x}_i} K_*^T K^{-1} \tag{9}$$

Since we only capture the variance in the set of shapes, when target object is not centered we must also recover 2D pose. Similar to [11], we differentiate $f(\Phi, I)$ w.r.t. 4 pose parameters λ_p (translation on x and y, scale and rotation) and then do a single one shot optimisation of both 2D pose and shape. The differentials are:

$$\frac{\partial f}{\partial \lambda_p} = \frac{\partial f}{\partial \Phi} \left(\frac{\partial \Phi}{\partial x} \frac{\partial x}{\partial \lambda_p} + \frac{\partial \Phi}{\partial y} \frac{\partial y}{\partial \lambda_p} \right) \tag{10}$$

where $\frac{\partial \Phi}{\partial x}$ and $\frac{\partial \Phi}{\partial y}$ are computed using finite differences and $\frac{\partial x}{\partial \lambda_p}$ and $\frac{\partial y}{\partial \lambda_p}$ follow trivially.

4. Results

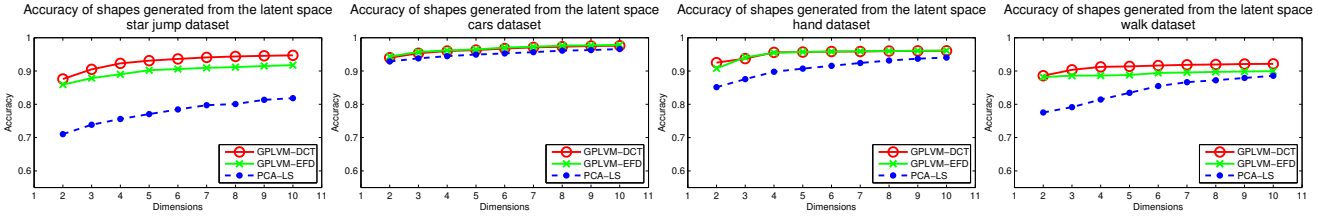

Figure 7. Accuracy comparison between our method (red), the standard one of using PCA on the embedding functions [15] (blue) and the elliptic Fourier one of [11] (green). We are constantly able to produce or better (or equal) results.

We tested our algorithm with several latent spaces and types of shape parameters. Average processing time per iteration was 210ms, with an unoptimised Matlab implementation, with convergence between 75 to 150 iterations.

We begin with a study of generative and convergence accuracy. Both accuracies are measured using area coverage.

Generative accuracy is tested by generating contours from various latent spaces, at several dimensionalities. In Figure 7 we compare the generative accuracy of 3 methods used to learn latent spaces of shapes: ours, the standard method of using PCA the embedding functions [15] and method of [11] where GP-LVM is used on the elliptic Fourier descriptors. We used the same latent spaces as [11]. In all cases our method is more accurate. The difference is especially big when the shapes are breaking up and merging (the star-jump and the walk). This is due to our use of the implicit shape representation (as opposed to [11]).

Convergence accuracy is tested by segmenting images for which we know the ground truth segmentation. Table 1 shows the convergence accuracy of our inference with and without the hierarchy of manifolds (using a 10D-4D-2D hierarchy). We learned shared spaces for a star-jump (with an example shape shown in Figure 6) and its corresponding pose, for a 2D car segmentation and its corresponding 3D model, and for a segmentation of a pair of eyes and the their corresponding fixation point on a computer screen. The results depend on the intrinsic level of non-linearity in the space (i.e. the more a space is nonlinear, the more PCA dimensions are required to capture all its variance). A more nonlinear manifold has more local minima so a higher dimensional space is more difficult to optimise in, as shown by the star jump example. The hierarchy helps explore the 10D space more extensively. When the manifold is more linear the hierarchy does not improve convergence accuracy by much. In all cases however, the hierarchy allows us to replace more costly 10D searches with less costly 2D and 4D searches, leading to average speedups of over 2.7x over running the optimisation in the 10D space alone (Table 2). This speedup is, of course, inverse proportional to the step size, but large step sizes can lead to divergence.

Next we apply our method to monocular 3D reconstruction. We learn joint latent spaces of car silhouettes (back and side views simultaneously) and 3D models (Figure 8). We use the same process to represent both 2D and 3D shapes: 2D DCT of the silhouette distance transform and 3D DCT of the distance transform of 3D volume boundary. A 3D model is generated as an isosurface of the zero level of the 3D distance transform generated from the shared space. Figure 9 shows our algorithm working, in spite of heavy occlusions, and correctly finding the modes in the latent space. The left panel shows an image with no ambiguity in the shape. All 3 mode searches converge to the same 2D shape and 3D model. The middle panel shows an ambiguous case where the rear of the car is occluded. The 3 searches yield putative segmentations and corresponding 3D models that range from sedan to hatchback, all equally likely given the silhouette data. Finally, in the right panel, there are no occlusions, but the back view of the car is still ambiguous. Here the 3 modes represent different car types: a small hatchback (first two modes) and a SUV (third mode).

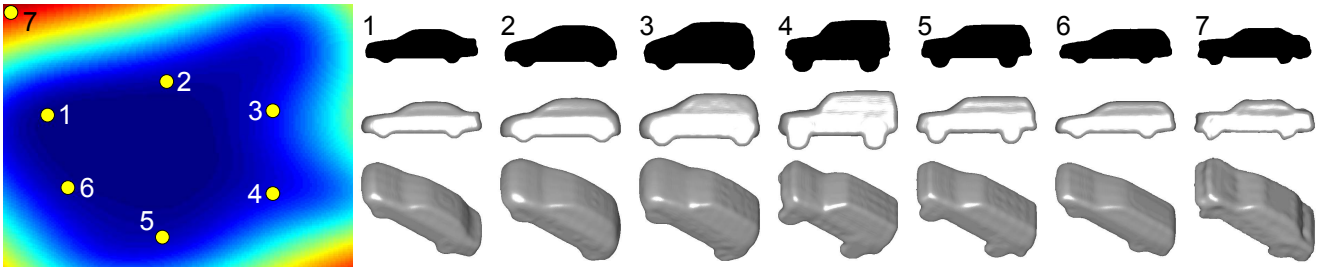

Figure 8. Example joint latent space (left) for side view car silhouettes and 3D models. Samples 1-6 are taken from a low variance region of the space (so more likely valid shapes) while sample 7 is taken from a high variance region (so less likely valid shapes).

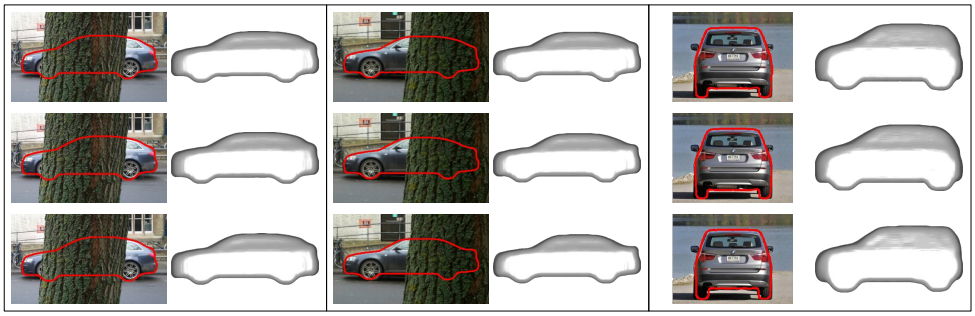

Figure 9. Examples of our algorithm correctly generating 3D car models from 2D car silhouettes.

Figure 10 shows our algorithm tracking gaze, by segmenting the sclera of the eye. We learn a joint latent space between the shape of the sclera and point of fixation on the screen (middle column, top). To get this position we displayed a grid of 16×10

dots (spaced at 32.23mm) on the screen (middle column, bottom) and looked towards each point. Images 5-12 show example training shapes. Their respective points of fixation are marked with green on the grid. Images 1-4 show segmentation results. Positions 1-4 on the grid show the ground truth (blue) and the value generated by our algorithm (red). Our method correctly and accurately segments all the parts of the sclera. The average error was 16.57mm or *1.58 degrees* visual angle error (the head was situated at around 60cm from the screen). While this error is bigger than the state of the art in gaze tracking with special lights and cameras, it is smaller than that of webcam-based gaze tracking (usually around 3 degrees).

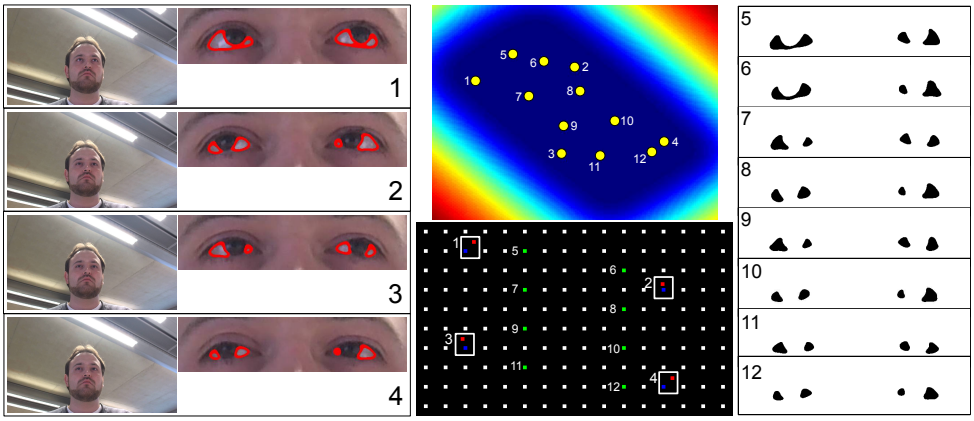

Figure 10. Gaze tracking. The latent space (middle column, top) generates both eye sclera shapes and location coordinates. To test and train we displayed a grid of dots (spaced at 32.23mm) on the screen (middle column, bottom) and looked towards each one. Images 5-12 show training shapes. Their respective screen locations are marked green on the grid. Images 1-4 show segmentation results. Positions 1-4 on the grid show the ground truth (blue) and the value generated by our algorithm (red).

Perhaps the most obvious application of our method is pose recovery. We therefore compared to two standard data sets, the Poser one of [1] and the HumanEva I one of [13]. Here we used 4D latent spaces, same as [4], but we did not use any latent space dynamics. The Poser data set consists of 1927 training and 418 test artificially generated images, from real mocap data. Table 3 shows that our method achieves state of the art results. Here, of the 3 modes, we selected the one that produced the highest image segmentation score. HumanEva I consists of video sequences with synchronised mocap data, recorded for 3 subjects. Figure 11 shows samples from the joint pose–shape space, for a walk in HumanEva I. Figure 12 shows results, on the same data set. The error in this data set is the average Euclidean distance between 15 3D joint locations, in camera coordinates. Since we do not recover 3D translation, we only provide results in local coordinates. This prevented us from computing the error on the actual test data from the data set (mocap information for this part is not included). We did test on the validation part of the data set (where mocap information is provided). The error was *28.10*mm, which would place our method close to state of the art in monocular 3D pose recovery. Dedicated pose recovery methods, with (marginally) better performance, do exist, such as [16], but we emphasise that this is only an application of our more general framework, that is not restricted to pose recovery.

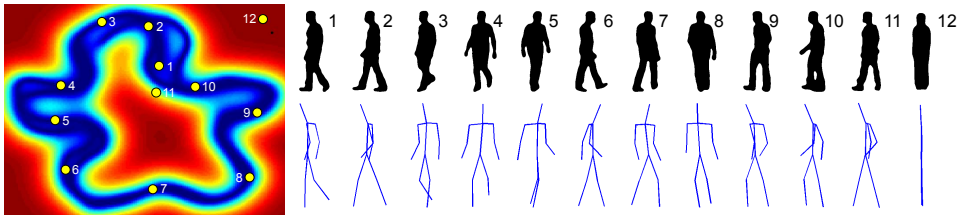

Figure 11. Example joint latent space (left) between 3D poses and 3D silhouettes for a walk from the HumanEva data set.

5. Conclusion

In this article we have proposed a method for simultaneous shape-constrained segmentation and parameter recovery. A classic example of this is monocular human pose recovery. In this application we have shown high quality segmentation and pose recovery that is at or beyond the current state of the art. We have also shown other example applications but we expect that the method has numerous other applications. In developing our system we have made three algorithmic

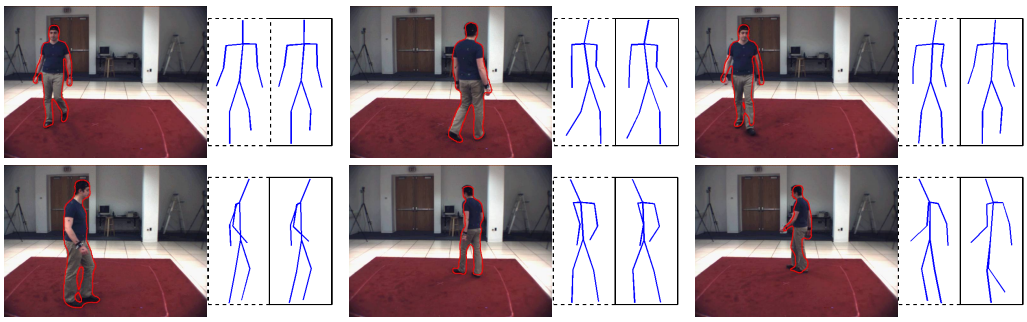

Figure 12. Example HumanEva results (4 accurate and 2 failure cases). Dotted line marks ground truth and filled line marks our results.

	10-4-2 Hierarchy	10D Alone	2D
Star jump	0.9373	0.9136	0.8438
Cars	0.9709	0.9676	0.9386
Eyes	0.8851	0.8837	0.8509

Table 1. Average convergence accuracy, with and without the hierarchy.

	10-4-2 Hierarchy	10D Alone	Speedup
Star jump	48.56	153.69	3.16
Cars	41.71	107.11	2.56
Eyes	20.67	50.49	2.44

Table 2. Average convergence speed (seconds) per frame, with and without the hierarchy.

Algorithm	RMS Angle Error (degrees)
Linear Regression [5]	7.70
Nearest Neighbour [14]	6.97
GPLVM [5]	6.50
Kernel Regression [14]	6.03
Gaussian RVM [1]	6.00
Global sKIE [14]	5.95
Local sKIE [14]	5.77
Shared Shape Spaces	5.25

Table 3. Performance of our algorithm on the Poser data set [1]

and representational contributions that may also have applications in other probe domains. First, we have shown how the multimodal space can be more effectively explored by introducing the shared shape – shape-similarity space. Second we have introduced a hierarchy of latent spaces in order to improve convergence and accuracy via a coarse to fine search. Finally, we have shown how shapes can effectively be represented implicitly as zero-level sets of an *approximation* to the signed distance transform embedding function, by compressing the embedding function using the 2D DCT. Our results demonstrate the efficacy of our representations, models and our inference algorithm.

A natural extension is to 3D; i.e. generate the 3D models as shown above and simultaneously optimise for both 3D model and 3D pose. Furthermore, unlike the explicit contour representation of [11], the extension to higher dimensions (eg for processing MRI data) is straightforward. Even though we have shown some results for video sequences (such as the walking and star-jump examples), our work is presently based on single images.

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. *CVPR*, 2004. 1, 8, 9

[2] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *ECCV*, pages 831–844, 2008. 6

[3] D. Cremers and G. Funka-lea. Dynamical statistical shape priors for level set based tracking. *T-PAMI*, 28:1262–1273, 2006. 1, 4

[4] C. H. Ek. *Shared Gaussian Process Latent Variable Models*. PhD thesis, Oxford Brookes University, 2009. 2, 8

[5] C. H. Ek, P. H. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In *MLMI*, volume LNCS 4892, pages 132–143, Jun. 2007. 1, 9

[6] S. A. Khayam. *The Discrete Cosine Transform (DCT): Theory and Application*. 2003. 5

[7] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *JMLR*, 6:1783–1816, 2005. 2

[8] N. D. Lawrence and J. Quinonero-Candela. Local distance preservation in the gp-lvm through back constraints. In *ICML*, pages 513–520, 2006. 2

[9] R. Navaratnam, A. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *CVPR*, pages 1–8, 2007. 1, 4

[10] M. Prasad. *Class-based Single View Reconstruction*. PhD thesis, University of Oxford, 2009. 1

[11] V. A. Prisacariu and I. Reid. Nonlinear shape manifolds as shape priors in level set segmentation and tracking. In *CVPR*, 2011. 1, 2, 4, 5, 6, 7, 9

[12] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. 2005. 2

[13] L. Sigal, A. Balan, and M. Black. HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. *IJCV*, 87:4–27, 2010. 8

[14] L. Sigal, R. Memisevic, and D. J. Fleet. Shared kernel information embedding for discriminative inference. *CVPR*, 2009. 9

[15] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, E. Grimson, and A. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *T-MI*, 22(2):137–154, 2003. 1, 4, 6, 7

[16] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *CVPR*, 2008. 8