The Joint Manifold Model for Semi-supervised Multi-valued Regression

Ramanan Navaratnam[1] Andrew W. Fitzgibbon[2] Roberto Cipolla[1]

[1]University of Cambridge
Trumpington St, Cambridge, UK
http://mi.eng.cam.ac.uk/{~rn246,~cipolla}

[2] Microsoft Research, Cambridge
7 JJ Thomson Ave, Cambridge, UK
http://www.research.microsoft.com/~awf

Abstract

Many computer vision tasks may be expressed as the problem of learning a mapping between image space and a parameter space. For example, in human body pose estimation, recent research has directly modelled the mapping from image features ($\mathbf{z}$) to joint angles ($\boldsymbol{\theta}$). Fitting such models requires training data in the form of labelled $(\mathbf{z}, \boldsymbol{\theta})$ pairs, from which are learned the conditional densities $p(\boldsymbol{\theta}|\mathbf{z})$. Inference is then simple: given test image features $\mathbf{z}$, the conditional $p(\boldsymbol{\theta}|\mathbf{z})$ is immediately computed. However large amounts of training data are required to fit the models, particularly in the case where the spaces are high dimensional.

We show how the use of unlabelled *data—samples from the marginal distributions $p(\mathbf{z})$ and $p(\boldsymbol{\theta})$—may be used to improve fitting. This is valuable because it is often significantly easier to obtain unlabelled than labelled samples. We use a Gaussian process latent variable model to learn the mapping from a shared latent low-dimensional manifold to the feature and parameter spaces. This extends existing approaches to (a) use unlabelled data, and (b) represent one-to-many mappings.*

Experiments on synthetic and real problems demonstrate how the use of unlabelled data improves over existing techniques. In our comparisons, we include existing approaches that are explicitly semi-supervised as well as those which implicitly make use of unlabelled examples.

1. Introduction

Many computer vision algorithms can be viewed as the design of a function which takes images as inputs and returns parameters of the imaged scene. In human body pose estimation, for example, the input to the function is a vector of image features, and the desired output is a probability density over the pose parameters of the human in the image. Much recent research [1, 7, 8, 15, 17, 21, 22] has adopted this "vision as regression" paradigm: given training examples comprising corresponding pairs of image features ($\mathbf{z}$) and joint angles ($\boldsymbol{\theta}$), learn a function $\boldsymbol{\theta} = f(\mathbf{z})$. This is an attractive paradigm because it promises fast and deterministic inference, loading most of the computational effort into the learning or regression phase.

This is a rather bare characterization of the paradigm, however, with a number of difficulties which are immediately apparent. First, in most cases of interest, the mapping between the pose space and image space can be many-to-many, so that f must be a one-to-many mapping. This is resolved by learning instead the conditional density, so that given an observed image with features $\mathbf{z}$, one can obtain a distribution over the pose, namely $p(\boldsymbol{\theta}|\mathbf{z})$.

The second difficulty, which we address in this paper, is that the dimensionality of the feature and pose spaces is typically relatively high, meaning that learning $p(\boldsymbol{\theta}|\mathbf{z})$ requires a considerable amount of labelled examples, i.e. corresponding $(\boldsymbol{\theta}, \mathbf{z})$ pairs. We show in this paper how to make use of *unlabelled* data to improve the estimate of the mapping. Unlabelled data are sets of pose parameters without corresponding images, as might be found in a motion capture database; or image features obtained from generic images of humans in motion.

This significantly reduces the number of training examples needed to learn complex mappings, meaning that applications which would previously have required too much labelled training data to be feasible are now possible.

1.1. Background

We combine a number of recent research results in order to achieve this. First we observe that most papers adopting the regression approach already *do* make use of unlabelled data, even though they may not mention semi-supervised learning. This is because many papers begin by projecting raw image features and pose parameters into a lower dimensional space before learning the mapping. Early work used principal components analysis [1], while more re-

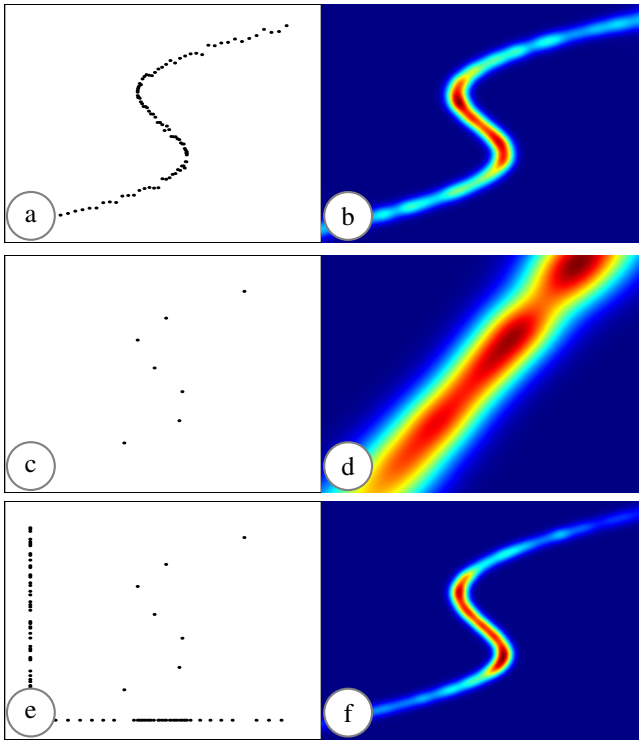


Figure 1. **Learning a manifold.** (a) 100 noisy samples from the curve $\{(t, t + 0.3\sin(2\pi t)) \mid 0 < t < 1\}$. (b) Joint density modelled by GPLVM. (c) 8 samples from the curve. (d) The manifold is not well modelled by the GPLVM. (e) The same 8 samples are augmented with 52 samples from each of the marginal distributions. (f) The marginal samples allow the manifold to be accurately estimated—note how the projection of (d) onto the horizontal axis does not match the samples on that axis in (e). See §1.2 for a fuller discussion.

cent techniques allow for nonlinear mappings [17, 21]. In terms of our notation, image features $\mathbf{z}$ are assumed to be images of latent points $\mathbf{t}_z$ on a low-dimensional manifold, parametrized by a projection function of the form $\mathbf{z} = P_z(\mathbf{t}_z)$, and these existing methods learn the function P_z. Similarly the pose parameters live in a low-dimensional manifold in joint-angle space parametrized by $\boldsymbol{\theta} = P_\theta(\mathbf{t}_\theta)$. Instead of learning the conditional $p(\boldsymbol{\theta}|\mathbf{z})$, which relates high-dimensional quantities, one instead learns the lower dimensional model $p(\mathbf{t}_\theta|\mathbf{t}_z) = f(\mathbf{t}_\theta, \mathbf{t}_z)$, from which $p(\boldsymbol{\theta}|\mathbf{z}) = f\big(P_\theta^{-1}(\boldsymbol{\theta}), P_z^{-1}(\mathbf{z})\big)$. The key is that the functions P_θ and P_z can be learned using all available image and pose samples, not just the those in the labelled set, so that the mapping is learned in a lower dimensional space, making better use of the labelled data.

The difficulty with this approach is that the $\mathbf{z}$ and $\boldsymbol{\theta}$ manifolds are learned independently, so that the function f which describes the mapping between them may be unnecessarily complex. In order to overcome this difficulty, *shared manifold* methods [7, 8, 15, 22] simultaneously learn the mappings P_z and P_θ from labelled samples, so that each $(\boldsymbol{\theta}, \mathbf{z})$ pair is associated with a single latent $\mathbf{t}$, giving $(\boldsymbol{\theta}, \mathbf{z}) = (P_\theta(\mathbf{t}), P_z(\mathbf{t}))$. These methods are, however, restricted to one-to-one mappings, and so cannot the model multimodal conditionals common in problems such as human pose estimation.

Closely related to our work is the study of *semi-supervised regression*, on which a survey can be found in [23]. Semi-supervised regression attempts to learn a mapping of the form $\mathbf{z} = f(\boldsymbol{\theta})$, using labelled pairs $(\boldsymbol{\theta}, \mathbf{z})$ and unlabelled $\boldsymbol{\theta}$ samples only. Our work improves on these methods by also incorporating samples from the $\mathbf{z}$ distribution, and by incorporating manifold learning to reduce the dimensionality of the joint space in which learning is performed.

One more related approach to the regression problem is direct modelling of the joint density $p(\mathbf{t}_\theta, \mathbf{t}_z)$, from which the conditionals are extracted using $p(\mathbf{t}_\theta|\mathbf{t}_z) = p(\mathbf{t}_\theta, \mathbf{t}_z)/p(\mathbf{t}_z)$. Agarwal et al.[2] propose fitting a mixture of Gaussians to the joint samples, and this was later extended [14] to incorporate missing data using the algorithm of Ghahramani and Jordan [5]. These approaches make use of unlabelled data, and can learn multimodal conditionals, but as we shall show in this paper, the mixture of Gaussians model needs many parameters, and hence a large number of labelled samples, in order to fit nonlinear manifolds.

Our primary modelling tool will be the Gaussian process latent variable model (GPLVM [11, 10]), which has already been used by several researchers in the context of human pose estimation [6, 15, 20, 21]. Our main contribution is to modify it to use marginal samples, and to demonstrate the improvement this confers.

1.2. The key idea

Figure 1 is an abstract illustration of the problem solved by this work. The horizontal axis represents images $\mathbf{z}$, and the vertical represents pose parameters $\boldsymbol{\theta}$. Given 100 training pairs $\{(\mathbf{z}_i, \boldsymbol{\theta}_i)\}_{i=1}^{100}$, we can get a good estimate of the joint density $p(\mathbf{z}, \boldsymbol{\theta})$, and consequently a good estimate of $p(\boldsymbol{\theta}|\mathbf{z})$ as shown in figure 1b. However in higher dimensions, samples are relatively more sparse, and we typically have a situation more akin to that in figure 1c, where only eight samples from the joint density are available. In this case, even the best algorithm we have (we show the results for lesser algorithms later in the paper) fails to model the manifold well. All is not lost, however: adding 104 *unlabelled* examples to the eight used in (d) allows the manifold to be reconstructed accurately. The unlabelled examples are simply samples from the marginal distributions $p(\boldsymbol{\theta})$ and $p(\mathbf{z})$.

This is a "pure" example of semi-supervised learning: because the features live in 1D spaces and the latent man-

ifold is 1D, there is no opportunity for manifold learning to leverage unlabelled data as described above. Without further prior knowledge, the smoothest mappings P_z and P_θ will be the identity. The unlabelled samples contribute to the estimate of the joint density because the marginals of the fitted joint density must match the distributions of $p(\mathbf{z})$ and $p(\boldsymbol{\theta})$. Put another way, knowing $p(\mathbf{z})$ gives a constraint on the joint $p(\boldsymbol{\theta}, \mathbf{z})$ of the form $\{\int p(\boldsymbol{\theta}, \mathbf{z})\mathrm{d}\boldsymbol{\theta} = p(\mathbf{z}), \forall \mathbf{z}\}$, and similarly, knowing $p(\boldsymbol{\theta})$ gives the constraint that $\int p(\boldsymbol{\theta}, \mathbf{z})\mathrm{d}\mathbf{z}$ must equal $p(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$.

The rest of this paper develops this idea for general regression problems, using the GPLVM as the manifold model. The final model is an assembly of various existing components, but the combination as implemented here is novel and, as we show, outperforms existing models on a number of real-world problems.

2. Joint Manifold Modelling (JMM)

Our ultimate goal is to model the joint density of $(\boldsymbol{\theta}, \mathbf{z})$. The model will depend on model parameters ϕ, and when we wish to make this clear, this density is written $p(\boldsymbol{\theta}, \mathbf{z}|\phi)$. A latent variable $\mathbf{t}$ drawn from distribution $p(t)$ generates each of $\boldsymbol{\theta}$ and $\mathbf{z}$ giving the factorization

$$p(\boldsymbol{\theta}, \mathbf{z}|\mathbf{t}) = p(\boldsymbol{\theta}|\mathbf{t})p(\mathbf{z}|\mathbf{t}) \tag{1}$$

from which the joint density may be obtained as

$$p(\boldsymbol{\theta}, \mathbf{z}) = \int p(\boldsymbol{\theta}|\mathbf{t})p(\mathbf{z}|\mathbf{t})p(\mathbf{t})\mathrm{d}\mathbf{t} \tag{2}$$

Each conditional $p(\cdot|\mathbf{t})$ will be defined by a Gaussian process, as described below. Given training examples $\mathbf{D} = \{(\boldsymbol{\theta}_l, \mathbf{z}_l)\}_{l=1}^L$, the manifold learning task will be to associate with each example a latent value $\mathbf{t}_l$ to maximize the posterior

$$\begin{aligned} p(\mathbf{t}_{1..L}|\mathbf{D}) &\propto p(\mathbf{D}|\mathbf{t}_{1..L})p(\mathbf{t}_{1..L}) && (3)\\ &= p(\boldsymbol{\theta}_{1..L}, \mathbf{z}_{1..L}|\mathbf{t}_{1..L})p(\mathbf{t}_{1..L}) && (4)\\ &= p(\boldsymbol{\theta}_{1..L}|\mathbf{t}_{1..L})p(\mathbf{z}_{1..L}|\mathbf{t}_{1..L})p(\mathbf{t}_{1..L}). && (5)\end{aligned}$$

We begin by describing Gaussian processes and the GPLVM, and then show how the above procedure may be modified to deal with unlabelled examples.

2.1. Gaussian Processes (GPs)

The conditionals take the form of a Gaussian process, a distribution over functions which, when conditioned on training data, produces a radial basis function approximation to the data, along with associated covariance estimates. The Gaussian process is an elegant and powerful way to model regression, and for full details the reader is encouraged to consult one of the many introductory treatments (e.g. [3, 12]). For our purposes, a brief adumbration of the properties will suffice. Consider a scalar function $x(t)$ of a scalar parameter t. Let $x(\cdot)$ be drawn from a zero-mean Gaussian process with kernel function $\kappa(\cdot, \cdot)$. Then for any (possibly infinite) vector of values $T = [t_i]_i$, the vector of corresponding function values $X = [x(t_i)]_i$ is Gaussian distributed with distribution

$$p(X|T) = \mathcal{N}(X|0, \kappa(T, T)) \tag{6}$$

where $\mathcal{N}$ is the standard multivariate Gaussian

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\big(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\big)$$

Conditioning a GP on training data may be viewed as producing another GP, with a new kernel and mean function, which we may loosely think of as "parameters" of the conditioned or fitted GP. A set of training examples is represented as pairs of column vectors $\{(\mathbf{x}_l, \mathbf{t}_l)\}_{l=1}^L$, where each $\mathbf{x}_l \in \mathbb{R}^d$, and $\mathbf{t}_l \in \mathbb{R}^s$. We wish to determine the mean function and kernel of the GP conditioned on the training data. Arrange the training vectors into the rows of matrices $\mathbf{X}_{\text{train}} = [\mathbf{x}_1, ..., \mathbf{x}_L]^\top$ and $\mathbf{T}_{\text{train}} = [\mathbf{t}_1, ..., \mathbf{t}_L]^\top$. The recipe for "fitting" the GP is as follows:

1. Compute the $L \times L$ Gram matrix $\mathbf{K}$ with ij^{th} element $\kappa(\mathbf{t}_i, \mathbf{t}_j)$. Our implementation used a Gaussian kernel of the form

$$\kappa(\mathbf{t}, \mathbf{t}') = \alpha \exp\left(-\frac{\beta}{2}\|\mathbf{t} - \mathbf{t}'\|^2\right) + \lambda \mathbf{t}^\top \mathbf{t}' \tag{7}$$

   We define ϕ to be vector of hyperparameters $\{\alpha, \beta, \lambda, \gamma\}$, where γ is a noise precision (see below).

2. Compute the $L \times d$ weight matrix $\mathbf{W} = \mathbf{K}^{-1}\mathbf{X}_{\text{train}}$.

3. Define the pair of functions

$$\mathbf{k}(\mathbf{t}) = \kappa(\mathbf{t}, \mathbf{T}_{\text{train}}) = [\kappa(\mathbf{t}, \mathbf{t}_1), ..., \kappa(\mathbf{t}, \mathbf{t}_L)]^\top, \tag{8}$$

$$\Sigma(\mathbf{t}, \mathbf{t}') = \kappa(\mathbf{t}, \mathbf{t}') - \mathbf{k}(\mathbf{t})^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{t}'). \tag{9}$$

Given the above definitions, the conditional density at a point $\mathbf{t}$ is then

$$p(\mathbf{x}|\mathbf{t}) = \mathcal{N}\left(\mathbf{x}|\mathbf{W}^\top \mathbf{k}(\mathbf{t}), \boldsymbol{\Sigma}(\mathbf{t}, \mathbf{t})\mathbf{I} + \gamma^{-1}\mathbf{I}\right) \tag{10}$$

where x_i is the i^{th} component of vector $\mathbf{x}$. Optimizing the likelihood of $\mathbf{X}_{\text{train}}$ over the hyperparameters ϕ is achieved using a nonlinear optimizer such as scaled conjugate gradient [10, 13].

2.2. GP Latent Variable Model (GPLVM)

We are now given training examples $\mathbf{X} \subset \mathbb{R}^d$ which are assumed to live in a low-dimensional submanifold of $\mathbb{R}^d$. The goal of the GPLVM to discover this submanifold, that is to associate with each $\mathbf{x}_l$ a latent variable $\mathbf{t}_l$ representing its coordinates in the manifold. We find $\mathbf{T}$ by maximizing the likelihood of the training examples. The likelihood of $\mathbf{X}$ given $\mathbf{T}$ and the hyperparameters ϕ is given by a Gaussian process as above,

$$p(\mathbf{X}|\mathbf{T},\phi) = \prod_{i=1}^{d} \mathcal{N}\left(\mathbf{X}_{:,i}\middle|\mathbf{0}, \kappa(\mathbf{T},\mathbf{T}) + \gamma^{-1}\mathbf{I}\right) \tag{11}$$

The notation $\mathbf{X}_{:,i}$ is used to indicate the $L \times 1$ column vector constructed from the i^{th} dimension of the data points. The prior over $\mathbf{T}$ is independently Gaussian on each $\mathbf{t}_l$:

$$p(\mathbf{T}) = \prod_{l=1}^{L} \mathcal{N}\left(\mathbf{t}_l|\mathbf{0},\mathbf{I}\right). \tag{12}$$

Thus the posterior is given by

$$p(\mathbf{T},\mathbf{X}|\phi) = p(\mathbf{X}|\mathbf{T},\phi)p(\mathbf{T}) \tag{13}$$

Again, a suitable optimization method can be used to find the optimal set of values $(\mathbf{T},\phi)$ that minimizes the negative log posterior $\mathbb{L} = -\log p(\mathbf{T},\mathbf{X}|\phi)$ for a given data set. Applying the GPLVM to our problem (5), we define $\mathbf{x}_l$ as the concatenation

$$\mathbf{x}_l = \begin{pmatrix} \boldsymbol{\theta}_l \\ \mathbf{z}_l \end{pmatrix} \tag{14}$$

and proceed as just described.

2.3. Accommodating unlabelled data

An advantage of the GPLVM is that it is relatively straightforward to incorporate unlabelled examples. Due to the separation along each dimension in equation (11), data with some missing dimensions can be accommodated in the model as explained below.

From (14), each data point $\mathbf{x}_k$ is composed of two components, which in this section we name pose component $\mathbf{x}_k^p$ and the feature component $\mathbf{x}_k^f$, of dimension P and F respectively, so $P + F = d$. Then $\mathbf{x}_k = [\mathbf{x}_k^{p\top}\mathbf{x}_k^{f\top}]^\top$. Without loss of generality, we may reorder the matrix $\mathbf{X}$ into three blocks, containing respectively the joint, f-marginal and p-marginal samples, so

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^{pf} \\ \mathbf{X}^{p*} \\ \mathbf{X}^{*f} \end{bmatrix}, \text{ with block sizes } \begin{bmatrix} L \times P & L \times F \\ M \times P & * \\ * & N \times F \end{bmatrix} \tag{15}$$

where $*$ indicates the missing data. Now we can reformulate the likelihood of $\mathbf{X}$ given $\mathbf{T}$ (11) as follows:

$$p\Big(\mathbf{X}\Big|\mathbf{T},\phi\Big) = \prod_{i=1}^{P+F} p\Big(\begin{bmatrix} \mathbf{X}^{pf} \\ \mathbf{X}^{p*} \\ \mathbf{X}^{*f} \end{bmatrix}_{:,i} \Big| \begin{bmatrix} \mathbf{T}^{pf} \\ \mathbf{T}^{p*} \\ \mathbf{T}^{*f} \end{bmatrix}, \phi\Big) \tag{16}$$

$$= \prod_{i=1}^{P} p\Big(\begin{bmatrix} \mathbf{X}^{pf} \\ \mathbf{X}^{p*} \end{bmatrix}_{:,i} \Big| \begin{bmatrix} \mathbf{T}^{pf} \\ \mathbf{T}^{p*} \end{bmatrix}, \phi\Big) \times \times \prod_{i=P+1}^{P+F} p\Big(\begin{bmatrix} \mathbf{X}^{pf} \\ \mathbf{X}^{*f} \end{bmatrix}_{:,i} \Big| \begin{bmatrix} \mathbf{T}^{pf} \\ \mathbf{T}^{*f} \end{bmatrix}, \phi\Big) \tag{17}$$

$$= \prod_{i=1}^{P} \mathcal{N}\left(\mathbf{X}^p_{:,i}\middle|\mathbf{0},\mathbf{K}^p + \gamma^{-1}\mathbf{I}\right) \times \times \prod_{i=P+1}^{P+F} \mathcal{N}\left(\mathbf{X}^f_{:,i}\middle|\mathbf{0},\mathbf{K}^f + \gamma^{-1}\mathbf{I}\right) \tag{18}$$

The notation $\mathbf{K}^p$ indicates the kernel matrix computed from the set of latent variables $\mathbf{T}^{pf}$ and $\mathbf{T}^{p*}$, and similarly $\mathbf{K}^f$ is computed from $\mathbf{T}^{pf}$ and $\mathbf{T}^{*f}$. The matrices $\mathbf{X}^p_{:,i}$ and $\mathbf{X}^f_{:,i}$ are defined as follows:

$$\mathbf{X}^p_{:,i} = \begin{bmatrix} \mathbf{X}^{pf} \\ \mathbf{X}^{p*} \end{bmatrix}_{:,i}, \mathbf{X}^f_{:,i} = \begin{bmatrix} \mathbf{X}^{pf} \\ \mathbf{X}^{*f} \end{bmatrix}_{:,i} \tag{19}$$

It is important to address the uncertainty in the latent coordinates of the unlabelled set. Otherwise, when there is a (relatively) large number of unlabelled data, the model tends to learn the manifolds in the marginal dimensions because the small number of latent points associated with the labelled data that captures the joint-space manifold contributes significantly less to the objective function. This usually does not represent the joint-space manifold because the relationship between the feature (marginal) dimensions and pose (marginal) dimensions is one-to-many. Hence, the prior on $\mathbf{T}$, defined in (12), is modified as follows:

$$p(\mathbf{T}) = \prod_{l=1}^{L} \mathcal{N}\left(\mathbf{t}_l|\mathbf{0},\mathbf{I}\right) \prod_{l=L+1}^{L+N+M} \mathcal{N}\left(\mathbf{t}_l|\mathbf{0},\eta\mathbf{I}\right) \tag{20}$$

Here, η (> 1) controls the uncertainty of the latent points associated with the unlabelled data. In all our experiments we set $\eta = 100$.

Fitting the GPLVM to a set containing unlabelled data is then a matter of maximizing the posterior (13) having substituted the appropriate definitions of $p(\mathbf{X}|\mathbf{T},\phi)p(\mathbf{T},\phi)$ from (18) and (20).

2.4. Backprojection

The GPLVM gives a convenient form for $p(\mathbf{x}|\mathbf{t})$, but the reverse model $p(\mathbf{t}|\mathbf{x})$ can be multimodal, and does not have a simple functional form. A latent space point $\mathbf{t}$ corresponding to a new data point $\mathbf{x}$ can be found by minimizing the

negative log joint-posterior over the novel data point and the associated latent point. However in practice, we found that it is sufficient to seek several modes by random starts within the latent space during the optimization of the objective function $\mathbb{L}$ and use the point estimates to obtain the likelihoods.

2.5. Summary: Using the JMM for pose estimation

To concretize the above discussion, we summarize the steps in using the JMM for pose estimation. The training data is a set of pairs $\{(\boldsymbol{\theta}_i, \mathbf{z}_i)\}_{i=1}^{L}$, and two sets of unlabelled samples $\{\boldsymbol{\theta}_j^*\}_{j=1}^{M}$ and $\{\mathbf{z}_k^*\}_{k=1}^{N}$. Fitting the GPLVM associates with each sample a latent parameter $\mathbf{t}$, which is stored with the training examples, along with the Gaussian process matrices, giving a representation of the form

$$\Phi = \left\{(\boldsymbol{\theta}_i, \mathbf{z}_i, \mathbf{t}_i)_{i=1}^{L}; (\boldsymbol{\theta}_j^*, \mathbf{t}'_j)_{j=1}^{M}; (\mathbf{z}_k^*, \mathbf{t}''_k)_{k=1}^{N}\right\} \qquad (21)$$

to which can be added the Gaussian process matrices $\mathbf{K}_\theta, \mathbf{W}_\theta$ and $\mathbf{K}_z, \mathbf{W}_z$. At inference time, given a new image feature vector $\mathbf{z}$, we find the latent values $\mathbf{t}$ which are local maxima of $p(\mathbf{t}|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{t})p(\mathbf{t})$. This can be seeded relatively efficiently by nearest neighbour lookup of the $\mathbf{z}_i$ followed by gradient-based optimization of $p(\mathbf{t}|\mathbf{z})$. For each mode, we then compute $p(\boldsymbol{\theta}|\mathbf{t})$.

Note that this does not give us $p(\boldsymbol{\theta}|\mathbf{z})$, the distribution of joint angles conditioned on the *image* $\mathbf{z}$: that should be computed using $p(\boldsymbol{\theta}|\mathbf{z}) = \int p(\boldsymbol{\theta}|\mathbf{t})p(\mathbf{t}|\mathbf{z})\mathrm{d}\mathbf{t}$. We do not know of a way to make this integral tractable, but in practice $p(\boldsymbol{\theta}|\operatorname{argmax}_{\mathbf{t}} p(\mathbf{t}|\mathbf{z}))$ is a good substitute.

3. Implementation

The above section describes the theoretical formulation of the joint manifold model, but of course certain details need to be attended to for a practical implementation.

The kernel parameters α and β need to be constrained to be positive. In our experiments this was enforced by reparameterizing as e.g. $\alpha = \ln(1 + \exp(\alpha'))$ and optimizing over α'. The dimension of the latent embedding can found from the residual variance similar to approach of Tenenbaum *et al*. [19].

Suitable **initialization** procedures should be employed to initialize the latent variables $\mathbf{t}$. In our implementation, we first run the Isomap algorithm [19] on the labelled samples to assign latent coordinates. Then for each unlabelled sample, say $(\boldsymbol{\theta}, *)$, the set of joint space neighbours $\{(\boldsymbol{\theta}_l, \mathbf{z}_l) \mid \|\boldsymbol{\theta}_l - \boldsymbol{\theta}\| < \varepsilon\}$ is computed, and a single element randomly chosen. Local linear interpolation performed using the neighbors of this chosen joint-space sample in the latent space yields the $\mathbf{t}$ assignment for $(\boldsymbol{\theta}, *)$. In multimodal areas of the density, this will incorrectly assign latent coordinates to a proportion of the unlabelled samples, but the problem is mitigated by the broad priors on the latent coordinates for unlabelled samples defined by η in (20).

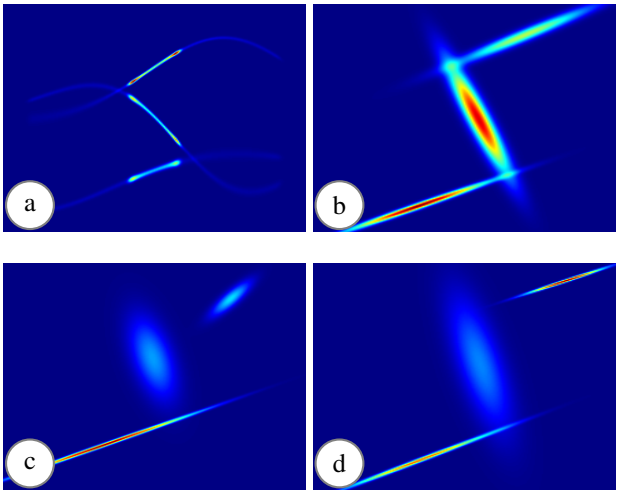


Figure 2. **Comparing methods.** (a) Mixture of experts (ME) fit to the S-curve with 15 points. The shape is tolerably approximated but not the density. ME cannot fit to fewer than 15 points on this data. (b) Gaussian mixture model fitted to 15 joint samples. (c) GMM on 8 samples. (d) GMM on 8 samples with 52 marginals. The GMM is a poorer fit to the manifold than the JMM (compare to figure 1), and gains less benefit from marginal data.

Although **temporal coherence** is not formally modelled in our approach, the video sequences were processed using a hidden Markov model (HMM). The HMM was fit to the proposed hypotheses and the Viterbi back-tracking algorithm was applied to produce some tracked video sequences. For each frame in the sequence, the multiple hypotheses proposed by the JMM were treated as the states and the confidence values obtained from the JMM as the state probabilities. The transition probabilities are computed from a nearest-neighbour dynamic model as follows. For each motion training sequence, a subset of frames are selected as keyframes. A transition matrix is learnt for these discrete key-poses from a large set of motion captured data. When applying the Viterbi algorithm, the transition probability between two poses is approximated by the transition probability between the two closest key-poses.

4. Experiments

Experiments were performed on synthetic and real examples to compare our approach (**JMM**) with existing algorithms. We compared to a selection of competing algorithms: nearest-neighbor (**NN**), mixture of experts (**ME**) models [1, 17], and joint density models (**GMM**) [2, 14]. We also implemented the shared latent structure technique of Shon *et al*. [15] and the manifold alignment of Ham *et al*. [7], denoted **S06** and **H06** respectively. Finally, we created

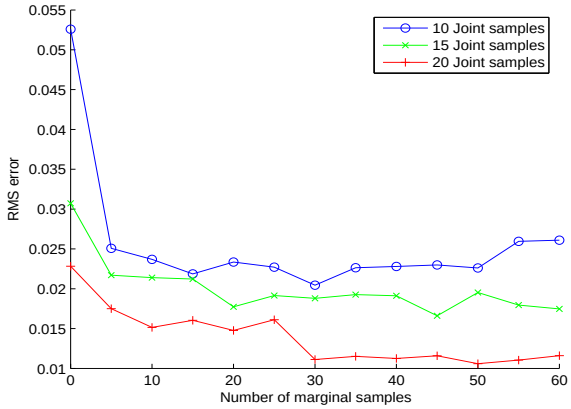


Figure 3. **Effect of number of marginal samples.** The S-curve (fig 1) is fit using the JMM, with varying numbers of joint and marginal samples.

missing-data versions of these two algorithms in a manner analogous to the missing data GPLVM, in order that they can use unlabelled data, which we call **S06*** and **H06***. Not all algorithms were tested on all sequences, generally because they were not appropriate, or could not run with the small numbers of joint samples we used. Only a subset of experiments are reported here; for more, and for video examples, see our website [4].

4.1. Synthetic problem

Experiments were carried out on the synthetic example from figure 1. Data points were samples from the curve $\{(\mathbf{z}, \boldsymbol{\theta}) = (t + 0.3\sin(2\pi t), t) \mid 0 < t < 1\}$ drawn from a uniform $p(t)$ to which were added Gaussian noise with zero mean and variance of 0.01. Thus the conditional $p(\boldsymbol{\theta}|\mathbf{z})$ is multivalued: for each $\mathbf{z}$ there may be up to three modes. Some $(\mathbf{z}, \boldsymbol{\theta})$ pairs are retained as the subset of points with known correspondence while the rest of the data are divided into separate collections of $(\mathbf{z}, *)$ and $(*, \boldsymbol{\theta})$ unlabelled points and treated as marginal data in algorithms for which this is possible. The JMM is fit as above.

Qualitative results of applying JMM, ME and GMM on 15 joint-samples are illustrated in figures 1 and 2. In general, JMM models the density better than either ME or GMM, and gains more benefit from marginal samples than GMM.

For the same toy problem, the performance of the JMM under varying combinations of number of joint-space and marginal samples was also investigated (figure 3). For each of 1000 test points sampled from the noise-free underlying manifold, the shortest distance to the predicted JMM mean function is computed. These distances were used in computing the RMS error. The results clearly show the benefit of introducing some marginal data. However, when a large number of marginal data is introduced, the error increases due to the effect described in section 2.3.

4.2. Human Body Pose Estimation

Experiments were conducted with real images and pose information obtained from motion capture systems [24]. Silhouette features were used as the basis to construct image descriptors. These image descriptors and the motion captured joint-angle information form our joint-space. Silhouette data for training (Fig.4) and testing were created using Poser^{TM}, a graphics package that can render realistic looking images from human models for given pose (joint-angle) information. Only these synthetically created silhouettes were used in the training stage of our approach, meaning that we could easily vary the ratio of labelled to unlabelled samples. Some experiments were performed using similar synthetic images that were not used in the training and motion captured on a different person. We also tested real images from video sequences of real humans performing a number of motions. We chose to use silhouettes as they yield one-to-many mappings.

The following subsections describe the experimental setup and the results obtained from applying our method on several types of motion. We proceed by explaining various error measures used to evaluate the results.

Interpreting Errors The error measures used during our tests can be classified into two, namely the error from the very first/most likely hypothesis (Type 1) and the minimum error between the first few hypotheses and the ground truth (Type 2). Each of these contain two subcategories. They are the average per-frame RMS error in pose (E_{pose}) and the average per-joint RMS angular error across frames (E_{angle}).

Given t frames of p dimensional ground truth pose data $\mathbf{T}$ and t hypotheses $\mathbf{H}$ corresponding to the test frames, these errors are defined as follows

$$E_{pose} = \frac{1}{t}\sum_{i=1}^{t}\sqrt{\frac{1}{p}\sum_{j=1}^{p}(T_{ij} - H_{i,j})^2} \qquad (22)$$

$$E_{angle} = \frac{1}{p}\sum_{j=1}^{p}\sqrt{\frac{1}{t}\sum_{i=1}^{t}(T_{ij} - H_{i,j})^2} \qquad (23)$$

Walking sequence The motion considered in this experiment is a person walking parallel to the image plane. The input image descriptor $\mathbf{z}$ is constructed from shape contexts of silhouette points. To work in a common coordinate system, these features are clustered into 40 clusters and each feature point is projected onto the common basis by weighted voting into the cluster centers [2, 16]. All the feature vectors for a silhouette are added to create a feature histogram to represent the image descriptor $\mathbf{z}$ for that

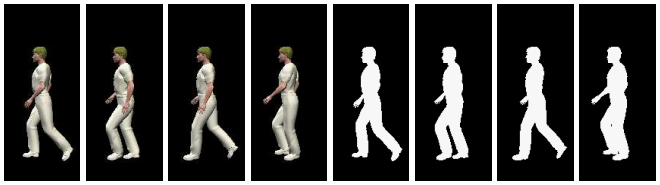
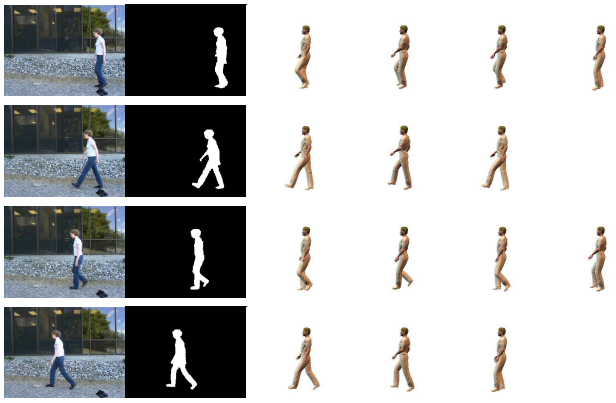


Figure 4. Samples of images and silhouettes used to train the JMM for a walking motion.



(a) (b) (c)

Figure 5. Results from applying a trained JMM for a walking motion. (a) Test images. (b) Silhouettes. (c) Modes of $p(\boldsymbol{\theta}|\mathbf{z})$ in decreasing order of likelihood. The left/right ambiguities arising from these silhouettes are correctly predicted by the JMM.

silhouette. The pose $\boldsymbol{\theta}$ is represented by a vector composed from 29 joint-angles. Since walking is a cyclic motion, a 2 dimensional latent space is used. The dimensionality is confirmed separately by looking at the residual variance on applying Isomap [19] on a larger data set. Two of the competing methods, GMM and ME, were not tested in this experiment as only small numbers (8, 16 and 32) of labelled samples were used.

Some results from real images are illustrated in Fig.5. This was obtained by training our model on 16 joint-samples only. It can be seen that both (left-right) flipped hypotheses were correctly identified. A comparison of our approach against other algorithms is tabulated in table 1. The test was repeated 5 times for 30 different images at each iteration. Our approach has consistently lower error and introducing unlabelled examples lowers the error further.

Golf swing sequence The motion considered in this experiment, from [21], is a person performing a golf swing facing the image plane. The input image descriptor $\mathbf{z}$ is constructed from the seven Hu moments [9]. In this case we do not have a background image, so the silhouettes were computed with some user interaction. Thus our results are not intended to imply that this image can be automatically processed, but do allow comparison of the various algorithms on this silhouette data.

Method	#J	#M*	Average RMS error			
			All hypotheses		First hypothesis	
			Pose	Angle	Pose	Angle
JMM	8	0	3.82	3.02	6.91	6.06
	8	8	3.68	2.90	6.78	5.90
	8	16	3.55	2.80	6.41	5.79
	8	32	3.83	3.04	8.05	6.67
	16	0	3.71	2.89	6.01	5.39
	16	16	3.60	2.71	5.88	5.17
	32	0	3.66	2.86	6.48	5.87
	32	32	3.45	2.65	6.41	5.79
NN	8	0	4.18	3.82	7.87	6.56
	16	0	3.95	3.63	6.70	5.87
	32	0	3.91	3.39	6.61	5.97
S06	8	0	-	-	8.63	7.11
	16	0	-	-	8.49	7.02
	32	0	-	-	7.48	6.93
H05	8	0	-	-	9.50	6.65
	16	0	-	-	7.63	5.39
	32	0	-	-	7.42	5.88
S06*	8	8	-	-	8.45	6.95
	8	16	-	-	8.28	6.81
	8	32	-	-	8.75	7.37
	32	32	-	-	7.01	6.59
H05*	8	8	-	-	8.49	6.21
	8	16	-	-	8.12	5.80
	8	32	-	-	7.75	5.37
	32	32	-	-	6.98	5.48

Table 1. **Walking sequence results.**

The pose $\boldsymbol{\theta}$ is represented by a vector composed from 29 joint-angles. Since this motion is unidirectional, only a single dimensional latent space is used. The test was repeated five times for 30 different images at each iteration. The average errors are tabulated in table 2. This was obtained by training our model on 16 joint-samples only. Since all but one of the seven Hu moments are invariant to reflection, some mirrored poses are predicted as one of the possible hypotheses although they accompanied by a low confidence on this prediction. The ME used 5 experts.

Exercise sequence An exercise routine, the results of which the reader is encouraged to consult at [4]. The sequence is rather more challenging than the previous examples as it is long, contains a variety of motions, and the silhouette extraction is made unreliable by wall markings the same colour as the subject. We assumed a 3 dimensional manifold for an exercise motion and applied JMM to learn the mapping. As can be seen in [4], a qualitatively successful track is obtained; the quantitative improvements in RMS pose or angle over S06, H06, and ME are at worst 18%, on average about 30%. Note that this is computed over the best hypothesis, as S06 and H06 cannot return multiple hypotheses.

Figure 6. Samples of images and silhouettes used to train the JMM for a golf swing motion.

Figure 7. Results from applying a trained JMM for a golf swing motion. Only 16 joint-space/labelled samples were used. Figures in the first row are the test images. The second row contains silhouettes of those test images. The rest of the rows contain hypotheses predicted by applying our method in the order of decreasing likelihood from top to bottom. The JMM has correctly identified the ambiguous situation and predicted accordingly.

Method	#J	#M*	Average RMS error			
			All hypotheses		First hypothesis	
			Pose	Angle	Pose	Angle
JMM	8	0	8.78	7.74	15.36	13.73
	8	16	6.77	5.78	11.16	10.07
	8	32	8.83	7.58	15.15	14.06
	16	0	7.22	6.94	14.16	13.42
	16	16	6.82	6.02	14.25	12.24
	32	0	6.98	6.40	12.26	10.65
	32	32	6.55	6.27	11.18	9.54
NN	8	0	10.34	9.64	17.29	16.02
	16	0	9.36	8.80	16.54	15.68
	32	0	9.18	8.60	14.16	13.99
S06(*)	8	0	-	-	15.80	14.50
	8	32	-	-	15.29	13.94
	32	0	-	-	12.09	11.51
	32	32	-	-	11.93	10.17
H05(*)	8	0	-	-	15.59	12.98
	8	32	-	-	11.23	10.21
	32	0	-	-	12.48	11.22
	32	32	-	-	11.32	10.35

Table 2. **Golf swing results**

Method	#J	#M*	Average RMS error			
			All hypotheses		First hypothesis	
			Pose	Angle	Pose	Angle
JMM	248	0	4.07	3.75	5.66	5.35
	248	248	3.60	3.64	5.02	4.75
NN	248	0	5.09	4.96	6.31	6.06
S06	248	0	-	-	7.49	6.77
	248	248	-	-	7.05	6.15
H06	248	0	-	-	9.48	8.25
	248	248	-	-	11.93	10.48
ME	248	0	8.92	7.70	10.15	9.54
	248	248	8.35	7.29	9.96	9.44

Table 3. **Exercise sequence results.**

5. Discussion

We have shown how the use of unlabelled examples in a vision-by-regression framework can improve the learned mapping. We know of no work that deals with this problem of semi-supervised learning of many-to-many mappings between feature spaces. We believe that the compelling results in figure 1 may apply even more strongly in higher dimensions, and that there is considerable potential for these techniques to expand the scope of the vision-by-regression paradigm.

On the other hand, the claims of semi-supervised methods are oft-heard, but the techniques not so often seen in practice. One reason may be our observation that most existing VBR techniques *are* semi-supervised, to the extent that they learn low-dimensional manifold representations before fitting a regressor. Therefore such systems should be considered semi-supervised methods. Our technique (and [8, 15, 22], albeit without unlabelled samples) improves on these because the jointly learned manifolds are forced to be in correspondence, while separately learned manifolds may be arbitrarily difficult to align post-hoc.

It should be noted that because our joint samples are obtained from a computer-graphic simulation, we do not really have the limitation that labelled examples are hard to obtain. However, this use of simulated training data (which is verified on real-world sequences) does allow the benefit of incorporating the unlabelled samples to be measured. We hope now to reap this benefit by addressing harder pose estimation problems, where the silhouette is not available, and natural image texture must be exploited instead.

Another criticism is that the model we propose does not at first sight appear to be a regressor. In order to find θ from z, one must first go through an optimization step, com-

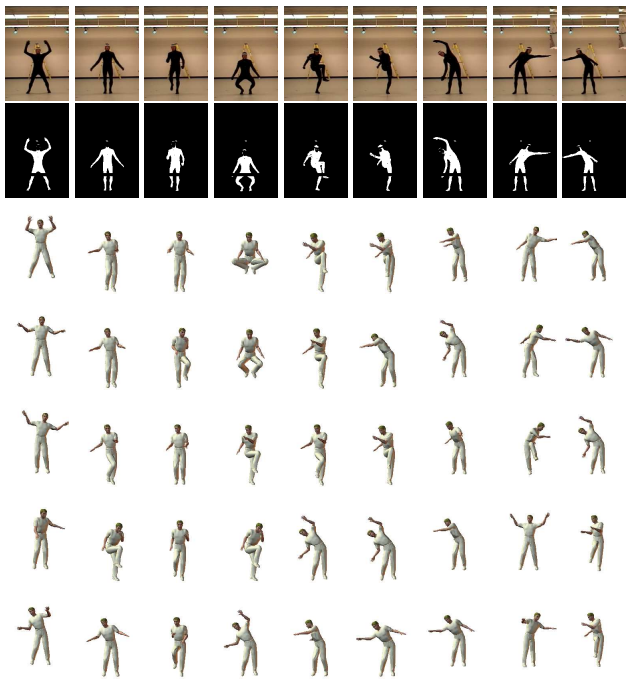


Figure 8. Results from applying a trained JMM for an exercise routine. Only 239 joint-sapce/labelled samples were used. Figures in the first row are the test images. The second row contains silhouettes of those test images. The rest of the rows contain hypotheses predicted by applying our method in the order of decreasing likelihood from top to bottom. Our approach has correctly identified the ambiguous situation and predicted accordingly.

puting $\min_t p(z|t)$ by gradient-based search. This could be mitigated by learning a one-to-many mapping from z to t, for example using ME. Although ME does not model the density particularly well, using it to seed a small number of initial estimates could yield an efficient implementation of the reverse model. Alternatively, we can be quite sure of finding all the modes by beginning the search at each radial basis function (RBF) centre $\mathbf{t}_i$. For RBFs with compact support (i.e. $k(\mathbf{t}, \mathbf{t}')$ goes quickly to zero as $|\mathbf{t} - \mathbf{t}'|$ increases), geometric data structures such as kd-trees can reduces this cost considerably, indeed becoming sublinear in the number of RBF centres.

We have not talked about efficiency in training and regression. However, promising results from active research in this area [18] might address this issue in future. Furthermore, the experiments have showed that the GPLVM can learn a reasonably good approximation of manifolds even with a small number of samples.

The tendency of additional marginal samples to ultimately worsen the result is commonly associated with semi-supervised techniques, and our introduction of η in (20) is a rather ad-hoc compensation for this effect. The explanation is that because the training samples are not actually drawn from a GPLVM, but from some distribution which we hope to approximate by a GPLVM, there will always be a "fitting error" which has different effects in the marginal and joint spaces. However, when manifold fitting is done separately, before learning the joint, it is often considered best to use as many samples as possible. We hope that future work might resolve this dichotomy.

Acknowledgements

This paper has benefitted from discussions with many people, to whom we are very grateful. Despite the risk of forgetting someone, we mention Nanthan Thayananthan, Ollie Williams, Chris Bishop, Mark Everingham, Ed Snelson, Tom Minka, Daniel Cremers, Andrew Zisserman, and Andrew Blake.

References

[1] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevant vector regression. In *CVPR*, 2004.

[2] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. In *IEEE Workshop on Vision for HCI at CVPR*, 2005.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] A. Fitzgibbon. *The Joint Manifold Model*, 2007. Website and technical report, http://www.research.microsoft.com/∼awf/jmm.

[5] Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an EM approach. In *Proc. NIPS*, volume 6, pages 120–127, 1994.

[6] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovi. Style-based inverse kinematics. *ACM Transactions on Graphics*, 23(3):522–531, 2004.

[7] J. Ham, D. Lee, and L. Saul. Semisupervised alignment of manifolds. In *AISTATS*, pages 120–127, 2005.

[8] J. H. Ham, I. Ahn, and D. Lee. Learning a manifold-constrained map between image sets: applications to matching and pose estimation. In *CVPR*, 2006.

[9] M. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, (2), 1962.

[10] N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. In *J. Machine Learning Res.*, pages 1783–1816, 2005.

[11] N. Lawrence and M. I. Jordan. Semi-supervised learning via Gaussian processes. In *Proc. NIPS*, 2005.

[12] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[13] A. F. Moller. A scaled conjugate gradient algorithm for fast supervised learning. In *Neural Networks*, volume 6, pages 525–533, 1993.

[14] R. Navaratnam, A. W. Fitzgibbon, and R. Cipolla. Semi-supervised learning of joint density models for human pose estimation. In *Proc. BMVC*, pages 679–688, Sept. 2006.

[15] A. P. Shon, K. Grochow, A. Hertzmann, and R. Rao. Learning shared latent structure for image synthesis and robotic imitation. In *Proc. NIPS*, pages 1233–1240, 2006.

[16] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3D human motion estimation. In *CVPR*, 2005.

[17] C. Sminchisescu, A. Kanaujia, Z. Li, and D. N. Metaxas. Conditional visual tracking in kernel space. In *Proc. NIPS*, 2005.

[18] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Proc. NIPS*, 2005.

[19] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. In *Science*, volume 290, pages 2319–2323, 2000.

[20] T. P. Tian, R. Li, and S. Sclaroff. Articulated pose estimation in a learned smooth space of feasible solutions. In *Proceedings of IEEE Workshop on Learning in Computer Vision and Pattern Recognition*, 2005.

[21] R. Urtasun, D. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *CVPR*, June 2006.

[22] J. J. Verbeek. Learning nonlinear image manifolds by global alignment of local linear models. *IEEE Trans. Pattern Analysis and Machine Intell.*, 28(8):1236–1250, 2006.

[23] X. Zhu. Semi-supervised learning literature survey. CS TR 1530, Univ of Wisconsin-Madison, 2005.

[24] http://mocap.cs.cmu.edu/.