Image and Vision Computing 31 (2013) 223–230

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Editor's choice article

Canonical locality preserving Latent Variable Model for discriminative pose inference☆

Yan Tian [a,b,*], Leonid Sigal [d], Fernando De la Torre [c], Yonghua Jia [a]

[a] Hangzhou Hikvision Digital Technology Co., Ltd, Hangzhou, P.R. China
[b] Zhejiang University, Hangzhou, PR China
[c] Carnegie Mellon University, Pittsburgh, USA
[d] Disney Research, Pittsburgh, USA

ARTICLE INFO

Article history:
Received 5 September 2011
Received in revised form 26 April 2012
Accepted 17 June 2012

Keywords:
Human pose estimation
Gaussian Mixture Regression
Latent Variable Model
Discriminative Model

ABSTRACT

Discriminative approaches for human pose estimation model the functional mapping, or conditional distribution, between image features and 3D poses. Learning such multi-modal models in high dimensional spaces, however, is challenging with limited training data; often resulting in over-fitting and poor generalization. To address these issues Latent Variable Models (LVMs) have been introduced. Shared LVMs learn a low dimensional representation of common causes that give rise to both the image features and the 3D pose. Discovering the shared manifold structure can, in itself, however, be challenging. In addition, shared LVM models are often non-parametric, requiring the model representation to be a function of the training set size. We present a parametric framework that addresses these shortcomings. In particular, we jointly learn latent spaces for both image features and 3D poses by maximizing the non-linear dependencies in the projected latent space, while preserving local structure in the original space; we then learn a multi-modal conditional density between these two low-dimensional spaces in the form of Gaussian Mixture Regression. With this model we can address the issue of over-fitting and generalization, since the data is denser in the learned latent space, as well as avoid the need for learning a shared manifold for the data. We quantitatively compare the performance of the proposed method to several state-of-the-art alternatives, and show that our method gives a competitive performance.

1. Introduction

Monocular pose estimation has been a focus of much research in computer vision due to abundance of applications for marker-less motion capture (MoCap) technologies. Marker-less MoCap spans a large number of application domains including entertainment, sport rehabilitation and training, activity recognition, human computer interaction and clinical analysis. Despite much research, however, monocular pose estimation remains a difficult task; challenges include high-dimensionality of the state space, image clutter, occlusion, lighting and appearance variation, to name a few.

Most prior methods can be classified into two classes of approaches: *generative* and *discriminative*. *Generative* approaches [1,2] define an image formation model by predicting appearance of the body **x** given a hypothesized state of the body (pose) **y**; an inference framework is then used to infer the posterior, $p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ over time. Since the inference often takes the form of non-convex search in a high-dimensional space of body articulations, these methods are computationally expensive and can suffer from local convergence (typically requiring a good initial guess for pose to seed the inference).

Discriminative approaches [3–16] avoid building an explicit imaging model, and instead opt to learn regression function, $\mathbf{y}=f(\mathbf{x})$, that maps from image features, **x**, to 3D poses, **y**; or probabilistically, a conditional distribution $p(\mathbf{y}|\mathbf{x})$ directly. The main goal is to learn a model from labeled training data, $\{\mathbf{x}^{(i)},\mathbf{y}^{(i)}\}_{i=1}^{N}$, that provides efficient and effective generalization for new examples at test time. The difficulty with this class of methods is two-fold: (1) the conditional probability of poses given image features, $p(\mathbf{y}|\mathbf{x})$, is typically multi-modal: different image features can be explained by several 3D poses; and (2) learning high dimensional regression functions, or conditional distributions, using limited training data is challenging and often results in over-fitting. Here we focus on discriminative pose estimation.

To deal with multi-modality, on the parametric side, mixture models were introduced, *e.g.*, Mixture of Regressors [4] or Mixture of Experts [14]. On the non-parametric side, local models that cluster data into convex sets and use uni-modal predictions within each cluster became popular (*e.g.*, Local Gaussian Process Latent Variable Models (Local GPLVM) [16]). In both cases over-fitting and generalization remained an issue, due to the need for large training datasets, as noted in [12] (Fig. 1).

☆ Editor's Choice Articles are invited and handled by a select rotating 12 member Editorial Board committee. This paper has been recommended for acceptance by Vladimir Pavlovic.

* Corresponding author at: Carnegie Mellon University, Pittsburgh, USA. Tel.: +1 412 425 2365.

0262-8856/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2012.06.009

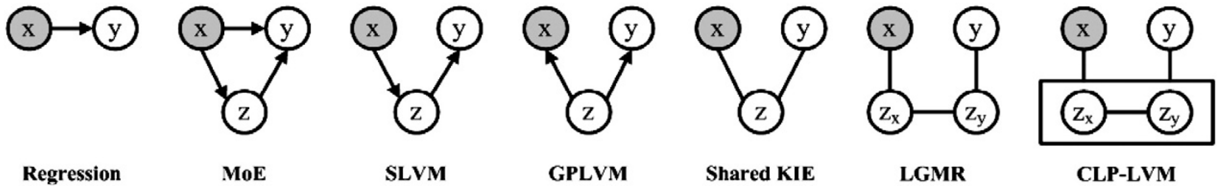

Fig. 1. Graphical model representations of models used for discriminative human pose estimation, including Regression Models [3,13], Mixture Models (*e.g.*, Mixture of Experts (MoE) [4,14]), Spectral Latent Variable Models (SLVM) [11], Gaussian Process Latent Variable Models [17,12], Shared Kernel Information Embeddings (sKIE) [18], Latent Gaussian Mixture Regression (Latent GMR) [19] and our Canonical Local Preserving Latent Variable Model. In all illustrations **x** denotes observed input variable corresponding to image features, **y** denotes the inferred 3D poses, and **z** corresponds to auxiliary latent variables (in case of Mixture of Experts (MoE) corresponding to the latent mixture component identity).

To alleviate the need for large labeled datasets Latent Variable Models (LVMs) were introduced as an intermediate representation. Kanaujia et al., [11], proposed Spectral LVMs to learn a non-linear latent embedding of the 3D pose data and a separately trained mixture model to map from the image features to the plausible latent positions in the sub-space. The relationship between the image features and latent space, however, was assumed to be linear within each mixture component.

Most traditional LVMs attempt to preserve distances between points in the original high-dimensional space. For example, if two human poses are close in the original space (according to some predefined distance metric), their latent representatives in the low-dimensional space should also be close and vice versa. Several latent variable techniques have been proposed to preserve the non-linear (or linear) structure of the high-dimensional data in the low dimensional space. He et al. proposed Locality Preserving Projections (LPP) to find the optimal linear approximation to the eigenfunctions of the Laplace Beltrami operator on the manifold [20]. Weinberger et al. presented maximum variance unfolding (MVU) which preserved distance by learning a kernel matrix [21]. Song et al. extended this method by Colored Maximum Variance Unfolding, and maximized the variance aligning with the side information (*e.g.* , labeled information), while preserving the local distance structures from the data [22]. However, all these methods are introduced in the context of learning a single low-dimensional representation for the data (*e.g.* , either image features or 3D poses, but not both); they contain no notion of input–output relationship between the features and poses that would facilitate discriminative inference.

As an alternative, Shared Gaussian Processes Latent Variable Model (Shared GPLVM) was introduced in [12] and [17], where the latent embedding was learned to preserve the joint structure of image features and 3D poses simultaneously; the forward non-linear mappings from the latent space to the input and output spaces were also learned at the same time. Due to the lack of backward mapping from the image features (and 3D poses) to the latent space, inference remained expensive, requiring multiple optimizations at the cost of $O(N^2)$, where N is the number of training examples. Shared Kernel Information Embeddings (sKIE) [18] provided closed form mappings to and from the latent space reducing the training and inference complexity by an order of magnitude. Shared GPLVM and sKIE are compelling, but are inherently non-parametric, with the model complexity being a function of the training set size; while this made them effective with small dataset it prevented their use with larger datasets. Alternatively, Supervised Local Subspace Learning (SL^2) [23] can learn directly a non-linear mapping from the feature space to the pose space without previously learning a joint latent space. SL^2 re-sample the pose space, and learn a mixture of local subspaces, one for each resampled point in the pose space. This feature makes SL^2 robust to non-uniform distribution of the feature space. Unfortunately, in our case, the pose space is high-dimensional and it will be computationally expensive to uniformly sample the input space (*i.e.* , pose space).

Dimensionality reduction for regression (DRR) techniques, instead of learning a joint embedding, opt for learning of a low-dimensional manifold embedding of the input data such that it preserves most, if not all, the necessary information for regression to the desired output. One way to formulate the DRR task is using the notion of sufficiency in dimension reduction (SDR) which find the subspace bases (or basis functions) such that the projected input yields the outputs independent of the original covariates. Manifold Kernel Dimensional Reduction (mKDR) presented in [24] is one such example, however, it involves a nonconvex optimization, potentially suffering from the existence of local minima. Alternatively, Covariance Operator Inverse Regression [25] generalizes Inverse Regression (IR) to nonlinear input/output spaces without explicit target slicing, but it assumes that the inverse regression is a smooth function.

We extend our work in [19], and propose Canonical Local Preserving Latent Variable Model (CLPLVM). Our formulation also extends [26], where traditional Canonical Correlation Analysis (CCA) was generalized to discover the low-dimensional manifold structure by maintaining the local information in the multiple data set. Similar to [26], we construct a cost function to find two sets of latent variables that keep local structure of the input image features and of the output 3D poses respectively, in their original high-dimensional spaces, while maximizing the correlation between related input and output latent variables at the same time.

Unlike [26], we also learn a multi-modal joint density model between the latent image features and the latent 3D poses, in the form of a Gaussian Mixture Model (GMM). GMM allows us to deal with multi-modality in the data and derive explicit conditional distributions for inference, in the form of Gaussian Mixture Regression (GMR).

2. Review of CCA and KCCA

2.1. Canonical Correlation Analysis (CCA)

CCA is a technique to extract common features from a pair of multivariate data. CCA, first proposed by Hotelling in 1936 [27], identifies relationships between two sets of variables by finding the linear combination of the variables in the first set (*e.g.* , image features) $\mathbf{X}=\{\mathbf{x}_1,\boldsymbol{x}_2,\ldots\mathbf{x}_N\}\in\mathbb{R}^{D\times N}$ (see notation[1]), that are most highly correlated with a linear combination of the variables in the second set $\mathbf{Y}=\{\mathbf{y}_1,\mathbf{y}_2,\ldots\mathbf{y}_N\}\in\mathbb{R}^{M\times N}$ (*e.g.* , 3D poses). N is the number of input–output points/pairs in our training dataset, D is the dimensionality of the image observations, and M is the dimensionality of the target 3D poses. Canonical Correlation Analysis solves for two projection matrices $\mathbf{B_x}\in\mathbb{R}^{D\times d_x}$ and $\mathbf{B_y}\in\mathbb{R}^{M\times d_y}$ which project the data into low-dimensional latent spaces that makes $\mathbf{X}$ and $\mathbf{Y}$ maximally correlated ($d_x<<D$ and $d_y<<M$). There exist several formulations for CCA, see

[1] Bold capital letters denote matrices (*e.g.*, **D**), bold lower-case letters represent column vectors (*e.g.*, **d**). All non-bold letters denote scalar variables. $\mathbf{d}_j$ is the j^{th} column of the matrix $\mathbf{D}$ .d_{ij} denotes the scalar in the i^{th} row and j^{th} column of **D**. $||\mathbf{d}||_2^2$ denotes the squared norm of the vector **d**. $Tr(\mathrm{A})=\Sigma_i a_{ii}$ is the trace of the matrix **A**. $\mathbf{D}=diag(\mathbf{a})$ is an operator that transforms a vector **a** into a diagonal matrix **D** such that $d_{ii}=a_i$. $\mathbf{I}_k$ denotes a $k\times k$ identity matrix.

[28] for a review. The projection matrices can be obtained by solving the following optimization problem [27,29,28]:

$$\begin{aligned} &min_{\mathbf{B_x},\mathbf{B_y}} - Tr\left(\mathbf{B_x^T}\mathbf{X}\mathbf{Y}^T\mathbf{B_y}\right) \\ &s.t.\quad \mathbf{B_x^T}\mathbf{X}\mathbf{X}^T\mathbf{B_x} = \mathbf{I} \\ &\qquad\ \ \mathbf{B_y^T}\mathbf{Y}\mathbf{Y}^T\mathbf{B_y} = \mathbf{I}. \end{aligned} \tag{1}$$

The constraints in Eq. (1) avoid the trivial solution of the projection matrices being unbounded and normalize the scale variance. Taking derivatives with respect to $\mathbf{B}_x$ and $\mathbf{B}_y$, it is easy to show that the critical points of the CCA correspond to the solutions of the following generalized eigenproblem (GEP):

$$\begin{pmatrix} 0 & \mathbf{X}\mathbf{Y}^T \\ \mathbf{Y}\mathbf{X}^T & 0 \end{pmatrix}\begin{pmatrix} \mathbf{b_x} \\ \mathbf{b_y} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{X}\mathbf{X}^T & 0 \\ 0 & \mathbf{Y}\mathbf{Y}^T \end{pmatrix}\begin{pmatrix} \mathbf{b_x} \\ \mathbf{b_y} \end{pmatrix}. \tag{2}$$

In regularized CCA [27,29], a regularization term $\gamma\mathbf{I}$, with $\gamma > 0$, is added to prevent over-fitting and avoid singularity. Specifically, regularized CCA solves the following generalized eigenvalue problem:

$$\begin{pmatrix} 0 & \mathbf{X}\mathbf{Y}^T \\ \mathbf{Y}\mathbf{X}^T & 0 \end{pmatrix}\begin{pmatrix} \mathbf{b_x} \\ \mathbf{b_y} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{X}\mathbf{X}^T + \gamma\mathbf{I} & 0 \\ 0 & \mathbf{Y}\mathbf{Y}^T + \gamma\mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{b_x} \\ \mathbf{b_y} \end{pmatrix}. \tag{3}$$

2.2. Kernel Canonical Correlation Analysis (KCCA)

One of the drawbacks of CCA is its limited ability to model non-linear dependencies between both sets that is essential in our application. Non-parametric KCCA [30,31] uses kernel methods to learn non-linear relations between two data sets without local minima (the resulting solution is a GEP).

Let $\mathbf{X}$ and $\mathbf{Y}$ be mapped into a Hilbert space through a non-linear mappings Φ and Ψ respectively. KCCA finds the (possible infinite dimensional) projection matrices $\mathbf{B}_{\Phi(\mathbf{x})}$ and $\mathbf{B}_{\Psi(\mathbf{y})}$ that maximize the correlation in the Hilbert space, that is:

$$\begin{aligned} &min_{\mathbf{B}_{\Phi(\mathbf{x})},\mathbf{B}_{\Psi(\mathbf{y})}} && -Tr\left(\mathbf{B}_{\Phi(\mathbf{x})}^T\Phi(\mathbf{X})\Psi(\mathbf{Y})^T\mathbf{B}_{\Psi(\mathbf{y})}\right) \\ &s.t. && \mathbf{B}_{\Phi(\mathbf{x})}^T\Phi(\mathbf{X})\Phi(\mathbf{X})^T\mathbf{B}_{\Phi(\mathbf{x})} = \mathbf{I} \\ &&& \mathbf{B}_{\Psi(\mathbf{y})}^T\Psi(\mathbf{Y})\Psi(\mathbf{Y})^T\mathbf{B}_{\Psi(\mathbf{y})} = \mathbf{I}. \end{aligned} \tag{4}$$

Making use of the representer theorem [32], the projection matrices can be expressed as a linear combination of the training samples, that is:

$$\begin{aligned} \mathbf{B}_{\Phi(\mathbf{x})} &= \Phi(\mathbf{X})\alpha_{\Phi(\mathbf{x})} \\ \mathbf{B}_{\Psi(\mathbf{y})} &= \Psi(\mathbf{Y})\beta_{\Psi(\mathbf{y})}. \end{aligned} \tag{5}$$

Using the previous expression, solving the KCCA problem is then equivalent to finding $\alpha_{\Phi(\mathbf{x})}$ and $\beta_{\Psi(\mathrm{y})}$, such that

$$\begin{aligned} &min_{\alpha_{\Phi(\mathbf{x})},\beta_{\Psi(\mathbf{y})}} && -Tr\left(\alpha_{\Phi(\mathbf{x})}^T\mathbf{K_x}\mathbf{K_y^T}\beta_{\Psi(\mathbf{y})}\right) \\ &s.t. && \alpha_{\Phi(\mathbf{x})}^T\mathbf{K_x}\mathbf{K_x}\alpha_{\Phi(\mathbf{x})} = \mathbf{I} \\ &&& \beta_{\Psi(\mathbf{y})}^T\mathbf{K_y}\mathbf{K_y^T}\beta_{\Psi(\mathbf{y})} = \mathbf{I}, \end{aligned} \tag{6}$$

where the Gram matrix $\mathbf{K_x} = \Phi(\mathbf{X})^T\Phi(\mathbf{X}), \mathbf{K_y} = \Psi(\mathbf{X})^T\Psi(\mathbf{X})$ can be expressed as a dot product in the Hilbert space. Similar to CCA, it is necessary to regularize the solution to avoid overfitting and rank-deficiency. The regularized KCCA problem has also a closed-form solution in terms of a GEP:

$$\begin{pmatrix} 0 & \mathbf{K_x}\mathbf{K_y} \\ \mathbf{K_y}\mathbf{K_x} & 0 \end{pmatrix}\begin{pmatrix} \alpha_{\Phi(\mathbf{x})} \\ \beta_{\Psi(\mathbf{y})} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{K_x}\mathbf{K_x} + \gamma\mathbf{I} & 0 \\ 0 & \mathbf{K_y}\mathbf{K_y} + \gamma\mathbf{I} \end{pmatrix}\begin{pmatrix} \alpha_{\Phi(\mathbf{x})} \\ \beta_{\Psi(\mathbf{y})} \end{pmatrix}. \tag{7}$$

3. Canonical Local Preserving Latent Variable Model

CCA models relation between the observation variables and the target variables in the linear latent spaces, while KCCA extends this relationship to non-linear latent spaces. As the *intrinsic* dimensionality of the data is typically much lower, regression between latent spaces found by CCA and KCCA tend to perform better than regression in the original space of image features and 3D poses. This is especially the case when the training data is limited, as will be illustrated in Section 5. However, both CCA and KCCA can be sensitive to lack of training data and regularized approaches introduce a prior in the latent space, that typically biases the solution. In this paper, we explore the use of a more natural generative regularizer. We propose Canonical Local Preserving Latent Variable Model (CLPLVM) that adds additional regularized terms that preserve local structure in the data while preserving appealing properties of CCA and KCCA.

Similarly to CCA, CLPLVM finds two projection matrices $\mathbf{B_x} \in \mathbb{R}^{D\times d_x}$ and $\mathbf{B_y} \in \mathbb{R}^{M\times d_y}$ such that the data in the two dimensionality-reduced spaces are maximally correlated. In doing so, however, it preserves *local* distances in both of those spaces. Moreover, we take multi-modality into consideration, by considering similarity between points (for distance preservation) only when their observation and target variables are both within local neighborhoods.

Formally, we formulate the CLPLVM model as the following optimization problem over projection matrices $\mathbf{B_x}$ and $\mathbf{B_y}$:

$$\begin{aligned} &min_{\mathbf{B_x},\mathbf{B_y}} && \theta_\mathbf{x}/2Tr\left(\mathbf{B_x^T}\mathbf{X}\mathbf{L_x}\mathbf{X}^T\mathbf{B_x}\right) + \theta_\mathbf{y}/2Tr\left(\mathbf{B_y^T}\mathbf{Y}\mathbf{L_y}\mathbf{Y}^T\mathbf{B_y}\right) - Tr\left(\mathbf{B_x^T}\mathbf{X}\mathbf{Y}^T\mathbf{B_y}\right) \\ &s.t. && \mathbf{B_x^T}\mathbf{X}\mathbf{X}^T\mathbf{B_x} = \mathbf{I} \\ &&& \mathbf{B_y^T}\mathbf{Y}\mathbf{Y}^T\mathbf{B_y} = \mathbf{I}, \end{aligned} \tag{8}$$

where the first two terms corresponds to Locality Preserving Projections [20] and are responsible for preserving the local distances in the two latent spaces. $\mathbf{L_x}$ and $\mathbf{L_y}$ are the Laplacian matrices from the adjacency matrices $\mathbf{W_x}$ and $\mathbf{W_y}$ given a neighborhood adjacency graph G. The last term ensures that the two latent spaces are maximally correlated. Consequently, $\theta_\mathbf{x}$ and $\theta_\mathbf{y}$ are weighting coefficients that control the contributions of these terms to the overall objective; A necessary condition for the minimum of the previous equation can be obtained taking derivatives with respect to $\mathbf{B_x}$ and $\mathbf{B_y}$. After some linear algebra, it can be shown that the solution corresponds to the following GEP:

$$\begin{pmatrix} \theta_\mathbf{x}\mathbf{X}\mathbf{L_x}\mathbf{X}^T & -\mathbf{X}\mathbf{Y}^T \\ -\mathbf{Y}\mathbf{X}^T & \theta_\mathbf{y}\mathbf{Y}\mathbf{L_y}\mathbf{Y}^T \end{pmatrix}\begin{pmatrix} \mathbf{B_x} \\ \mathbf{B_y} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{X}\mathbf{X}^T & 0 \\ 0 & \mathbf{Y}\mathbf{Y}^T \end{pmatrix}\begin{pmatrix} \mathbf{B_x} \\ \mathbf{B_y} \end{pmatrix}. \tag{9}$$

The only thing remaining to learn CLPLVM is the specification of the adjacency graph and the corresponding weights.

3.1. Construction of the adjacency graph

Let G denotes a graph with N nodes, where N is the number of input–output training pairs corresponding to image features and 3D poses: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_N\}$. The adjacency graph should tell us for every input–output pair $\{\mathbf{x}_i, \mathbf{y}_i\}$ a set of training pairs that are within I's neighborhood. Note that our method is different from [20], as an edge between nodes i and j is added if both $\mathbf{x}_i - \mathbf{x}_j$ and $\mathbf{y}_i - \mathbf{y}_j$ are close. We considered two variations:

- *ε-neighborhoods*. Nodes i and j are connected by an edge if $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 < \varepsilon$, where ε is the user specified constant related to the desired neighborhood size.
- *k nearest neighbors*. Nodes i and j are connected by an edge if $\mathbf{x}_i$ is among k nearest neighbors of $\mathbf{x}_j$ and $\mathbf{y}_i$ is among k nearest neighbors of $\mathbf{y}_j$ according to Euclidean distance in the two spaces.

In practice, we found k nearest neighbor method to be efficient and work well. Hence, this is the method we used in the remainder of the paper, including all the experiments. For all experiments we used $k = 12$ as the neighborhood size.

3.2. Computing the weights

Here, we also have two potential variants for the weighting of the edges. $\mathbf{W_x}$ and $\mathbf{W_y}$ are sparse symmetric $N \times N$ matrices with elements $\mathbf{W}_{\mathbf{x},ij}/\mathbf{W}_{\varnothing ut,ij}$ having the weight of the edge joining vertices i and j in the corresponding two spaces, and 0 if there is no such edge.

- *Heat kernel.* If nodes i and j are connected, then $\mathbf{W}_{\mathbf{x},ij} = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{t_x}}$, $\mathbf{W}_{\mathbf{y},ij} = e^{\frac{-\|\mathbf{y}_i - \mathbf{y}_j\|_2^2}{t_y}}$, otherwise $\mathbf{W}_{\mathbf{x},ij} = 0$, $\mathbf{W}_{\mathbf{y},ij} = 0$.
- *0/1 kernel.* $\mathbf{W}_{\mathbf{x},ij} = 1$ and $\mathbf{W}_{\mathbf{y},ij} = 1$ if and only if vertices i and j are connected by an edge.

Notice that with 0/1 kernel $\mathbf{W}_x = \mathbf{W}_y$. We found 0/1 kernel to be less effective than the heat kernel because it binarizes the distances. Hence, we use the heat kernel method for the weight assignment in the remainder of the paper, including all the experiments (the kernel parameters, chosen by cross validation, are $t_x = 10$ and $t_y = 1$).

The CLPLVM model computes closed-form linear mappings for the input features $\mathbf{x}$ and output 3D poses $\mathbf{y}$ to their corresponding latent spaces, *e.g.*, $\mathbf{z}_x = \mathbf{B^T x^X}$ and $\mathbf{z}_y = \mathbf{B^T y^Y}$, where $\mathbf{z}_x \ \mathbb{R}^{d}{}_{\mathrm{x}}$, $\mathbf{z}_y \ \mathbb{R}^{d}{}_{\mathrm{y}}$. However, 3D pose inference, in addition, requires a mapping or a distribution between $\mathbf{z}_x$ and $\mathbf{z}_y$. If one assumes a unimodal relationship between the two latent spaces, linear regression or a joint Gaussian density are reasonable choices as the two spaces are, by formulation, maximally correlated. In human pose inference, however, this is not the case and the mapping can and is, in general, multi-modal. Various forms of non-parametric regression can be used as an alternative. However, the complexity of non-parametric methods is typically high, being a function of the training set size and making the model hard to scale to large datasets. As a consequence, we propose to use a parametric Gaussian Mixture Regression instead.

4. Gaussian Mixture Regression

Given a latent observation vector, $\mathbf{z}_x \ \mathbb{R}^{d}{}_{\mathrm{x}}$, and the corresponding latent 3D pose, $\mathbf{z}_y \ \mathbb{R}^{d}{}_{\mathrm{y}}$, we assume the joint latent sample, $(\mathbf{z}_x, \mathbf{z}_y)$, follows the Gaussian mixture distribution with K mixture components,

$$p\left(\mathbf{z}_x, \mathbf{z}_y\right) = \sum_{k=1}^{K} \pi_k p\left(\mathbf{z}_x, \mathbf{z}_y; \mu_k, \Lambda_k\right) \tag{10}$$

where $\mathrm{p}(\mathbf{z}_x, \mathbf{z}_y; \mu_k, \Lambda_k)$ is the multivariate Gaussian density function. The parameters of model include prior weights, π_k, means, $\mu_k = \left[\mu_{k,\mathbf{z}_x} \ \ \mu_{k,nt}\right]^T$, and variances, $\Lambda_k = \left[\Lambda_{k,\mathbf{z}_x} \ \ \Lambda_{k,\mathbf{z}_x\mathbf{z}_y}; \Lambda_{k,\mathbf{z}_y\mathbf{z}_x} \ \ \Lambda_{k,\mathbf{z}_y}\right]$, of each Gaussian component.

The joint density can be expressed as the sum of the products of the marginal density of $\mathbf{z}_x$, and the probability density function of $\mathbf{z_y}$ conditioned on $\mathbf{z}_x$:

$$p\left(\mathbf{z}_x, \mathbf{z}_y\right) = \sum_{k=1}^{K} \pi_k p\left(\mathbf{z}_y \middle| \mathbf{z}_x; m_k, \sigma_k^2\right) p\left(\mathbf{z}_x; \mu_{k,\mathbf{z}_x}, \Lambda_{k,\mathbf{z}_x}\right). \tag{11}$$

Similarly, the marginal distribution,

$$p(\mathbf{z}_x) = \sum_{\mathbf{z}_y} p\left(\mathbf{z}_x, \mathbf{z}_y\right) = \sum_{k=1}^{K} \pi_k p\left(\mathbf{z}_x; \mu_{k,\mathbf{z}_x}, \Lambda_{k,\mathbf{z}_x}\right), \tag{12}$$

is also a mixture.

The global regression function can be obtained by combining Eqs. (11) and (12):

$$p\left(\mathbf{z}_y \middle| \mathbf{z}_x\right) = \frac{p\left(\mathbf{z}_x, \mathbf{z}_y\right)}{p(\mathbf{z}_x)} = \frac{\sum_{k=1}^{K} \pi_k p\left(\mathbf{z}_x; \mu_{k,\mathbf{z}_x}, \Lambda_{k,\mathbf{z}_x}\right) p\left(\mathbf{z}_y \middle| \mathbf{z}_x; m_k, \sigma_k^2\right)}{\sum_{k=1}^{K} \pi_k p\left(\mathbf{z}_x; thbf\mu_{k,\mathbf{z}_x}, \Lambda_{k,\mathbf{z}_x}\right)} \tag{13}$$

This can be expressed as a mixture of conditional distributions, $p(\mathbf{z}_y|\mathbf{z}_x) = \sum_{k=1}^{K} \omega_k p(\mathbf{z}_y|\mathbf{z}_x; m_k, \sigma_k^2)$, where the mixing weights ω_k are defined as:

$$\omega_k = \frac{\pi_k p\left(\mathbf{z}_x; \mu_{k,\mathbf{z}_x}, \Lambda_{k,\mathbf{z}_x}\right)}{\sum_{j=1}^{K} \pi_j p\left(\mathbf{z}_x; \mu_{j,\mathbf{z}_x}, \Lambda_{j,\mathbf{z}_x}\right)}. \tag{14}$$

The mean and the variance of the conditional distribution $p(\mathbf{z}_y|\mathbf{z}_x)$ can be acquired in closed form by:

$$m_k = \mu_{k,\mathbf{z}_x} + \Lambda_{k,\mathbf{z}_y\mathbf{z}_x} \Lambda_{k,\mathbf{z}_x}^{-1} \left(\mathbf{z}_x - \mu_{k,\mathbf{z}_x}\right) \tag{15}$$

$$\sigma_k^2 = \Lambda_{k,\mathbf{z}_y} - \Lambda_{k,\mathbf{z}_y\mathbf{z}_x} \Lambda_{k,\mathbf{z}_x}^{-1} \Lambda_{k,\mathbf{z}_x\mathbf{z}_y}. \tag{16}$$

The learning can be achieved with a simple Gaussian Mixture Model, using Expectation Maximization (EM) procedure with K-means initialization. The prediction given a new input can be obtained by computing expectation over $p(\mathbf{z}_y|\mathbf{z}_x)$:

$$\mathbf{z}_y^* = E_{p\left(\mathbf{z}_y|\mathbf{z}_x\right)}\left[\mathbf{z}_y\right] = \sum_{k=1}^{K} \omega_k m_k. \tag{17}$$

Alternatively, if the conditional relationship is truly multi-modal, it is better to look at the modes given by m_j directly. In general, we can have up to K distinct modes in the conditional distribution for a given input, $\mathbf{z}_x$.

4.1. Relationship to other methods

Notice that the regression function (17) derived from the joint mixture Gaussian density is of the form of a kernel estimator. However, there is a key difference with non-parametric regression: the mixture weights, ω_k, are not determined by the local structure of the data, but rather by the components of a global Gaussian mixture model.

The Nadaraya–Watson kernel smoother [33] is a Gaussian Mixture Regression model with $K = N$ components, where N is the total number of training points. At the other end of the spectrum, $K = 1$ is approximately the classical linear regression model. Hence, the Gaussian Mixture Regression model can, in principal, represent a spectrum of regression models, ranging from the non-parametric kernel regression, where $K = N$, to the classical linear regression, $K = 1$.

5. Experiment

We tested the performance of our method on three datasets: (1) Poser dataset – consisting of synthetic sequences produced by Poser software [34], (2) CMU dataset – comprising real motion capture dataset and video publicly available from [35], and (3) standard dataset with provided error metrics made available by Agarwal and Triggs [3].

5.1. Poser dataset

We synthesized image data from motion capture sequences using Poser 7 software. The motion sequences came from 8 categories: walk, run, dance, fall, prone, sit, transitions and misc (see Fig. 2). A total of 5 sequences within each category were broken into: 3 training

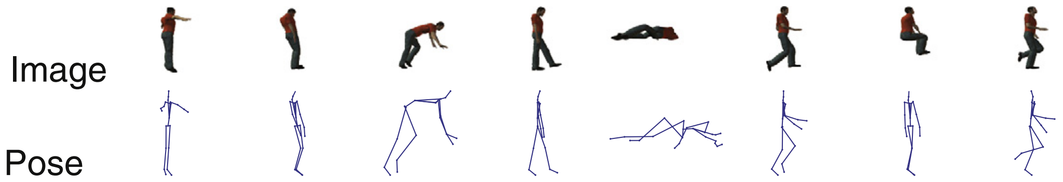

Fig. 2. Synthesized data generated by Poser 7 software.

Table 1
Evaluation of different algorithms on the Poser data set (for details see text).

Error (cm)		LR	PCA	LPP	LGMR [19]	CCA	KCCA	LPCCA [26]	CLPLVM	CLPLVM + GMR
Dance	S1	10.82	10.43	9.72	9.33	9.60	9.42	9.40	9.32	**9.11**
	S2	10.27	9.81	9.54	9.12	9.41	9.23	9.22	9.14	**8.78**
Falls	S1	12.32	11.80	11.15	10.74	10.95	10.90	10.88	10.72	**10.27**
	S2	12.31	11.70	11.06	10.57	10.82	10.80	10.77	10.65	**10.28**
Miscs	S1	8.32	8.59	8.42	8.00	8.12	8.03	8.00	7.97	**7.84**
	S2	12.27	12.19	12.10	11.71	11.80	11.74	11.71	11.63	**11.14**
Prone	S1	11.26	10.85	10.10	9.72	10.08	10.03	10.00	9.87	**9.43**
	S2	11.46	10.96	10.28	9.89	10.19	10.15	10.11	9.92	**9.61**
Run	S1	8.93	8.70	8.64	8.33	8.31	8.26	8.24	8.15	**7.97**
	S2	11.64	10.96	10.79	10.32	10.62	10.53	10.51	10.46	**10.04**
Sit	S1	19.32	19.15	19.09	**18.85**	19.03	18.99	18.97	18.97	18.89
	S2	12.33	12.25	12.23	**12.04**	12.16	12.11	12.11	12.12	**12.04**
Transition	S1	8.71	8.53	8.48	8.31	8.43	8.39	8.38	8.32	**8.27**
	S2	9.64	9.51	9.38	9.25	9.35	9.33	9.32	9.28	**9.22**
Walk	S1	11.64	11.16	10.66	10.16	10.33	10.28	10.27	10.22	**9.90**
	S2	9.16	8.75	8.44	8.06	8.15	8.12	8.11	8.13	**8.02**
Average		11.28	10.96	10.63	10.27	10.45	10.39	10.38	10.30	**10.05**

Bold data give the best performance.

and 2 testing sequences, with each sequence containing approximately 500 frames. The size of each synthetic image was 500 × 490 pixels. We represented body pose in terms of exponential map [36] of 23 joints, resulting in $M = 72$. All poses were represented relative to the skeleton root (pelvis).

5.1.1. Image features

We relied on silhouette features and encode them using vector quantized histograms of shape context features [37]. (The dimensionality of the resulting feature vector is $D = 100$).

5.1.2. Error measure

We used a standard average joint position error in line with prior works. We reported RMSE of average joint error in centimeters (*cm*).

We compared our CLPLVM model with a number of dimensionality reduction and regression alternatives, including: Principal Component Analysis (PCA), Locality Preserving Projections (LPP), Canonical Correlation Analysis (CCA), Kernel Canonical Correlation Analysis (KCCA), and Locality Preserving Canonical Correlation Analysis (LPCCA) [26], and the Linear Regression (LR) in the original high-dimensional space, as well as our earlier Latent GMR model (LGMR) in [19]. For CLPLVM, PCA, LPP, CCA, KCCA, LPCCA we used linear regression to learn a function between respective $\mathbf{z}_x$ and $\mathbf{z}_y$ for inference. We also gave the performance of our approach extended with Mixture Gaussian Regression (CLPLVM + GMR) as was proposed in Section 4. The results are shown in Table 1. All the parameters were leaned by cross-validation: *e.g.*, the width of the RBF kernel in KCCA was 0.5, mixture models had $K = 8$ components, and the PCA and LPP were trained to keep 95% of the original energy. The results for [26] in Table 1 are based on re-implementations of the original work.[2] In all cases we compared the *expectations* computed under the models (mean error) with ground truth; for CLPLVM + GMR this amounts to Eq. (17).

It is worth noting that [19] used different representation for the image features (60D global shape context representation as opposed to vector quantized histograms of shape context here) and a different representation of the pose (in terms of 3D joint positions as opposed to exponential map of joint angles), so results are not directly comparable.

5.2. CMU dataset

From CMU Graphics Lab Motion Capture Database, we chose sequences 17–01 to 17–05 as training data, and used sequences 17–07 to 17–09 as test data. The size of each image was 240 × 352 pixels. We represented body pose in terms of exponential map [36], resulting in $M = 96$ (CMU skeleton contains more joints than skeleton used by Poser above). The image features are implemented as before and have dimensionality of $D = 100$.

The images from CMU dataset are visualized in the 2D latent space in Fig. (3). We plotted 800 images in a random order, and the 100th frame and its successor were draw with the same color, so as the 200th, 300th, 400th, 500th, and 600th frames. Frames nearby had high similarities because they were captured at 30 frame/s. However, we could see that these contiguous samples were projected to distant points in the latent space using CCA. However, by combining the local distance preservation requirement into the learning, our model did a better job preserving the distance between samples, that were closer in the high dimensional space, in the low-dimensional latent embedding. Using our proposed method, samples that were alike in the high-dimensional space were more likely to cluster together in the latent space. Moreover, because of the extra constraints our method was more robust to lack of training samples.

We also compared our method with a number of alternatives using different sizes of training set, which could be seen in Table 2. Similar to Table 1, by default all methods utilize linear regression to map between the image feature and 3D pose latent spaces, except our earlier LGMR model [19] which use Gaussian Mixture Regression to find this mapping. Limited training data often results in over-fitting

[2] For the purpose of comparison, we did not explore the temporal prior which is employed in [10].

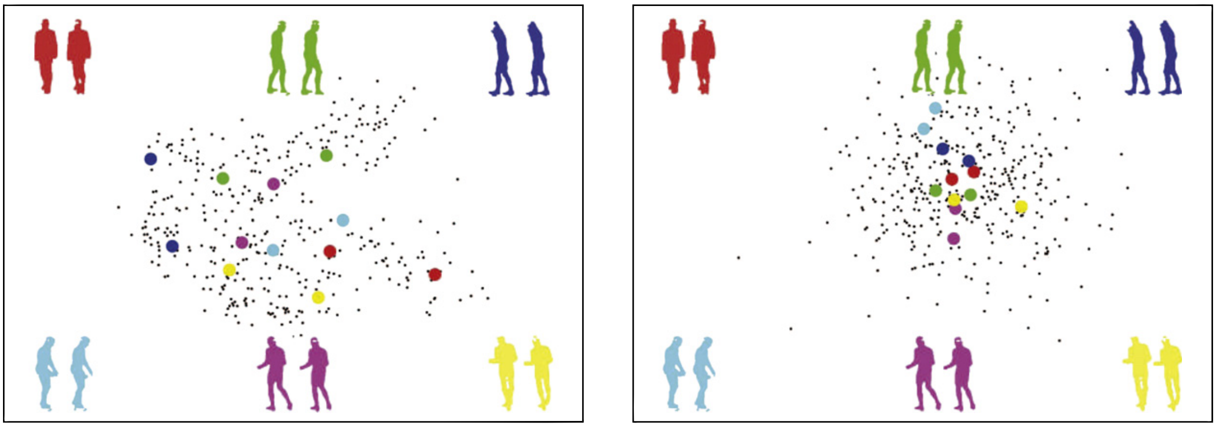

Fig. 3. Image visualization in the 2D latent space. Left: Latent space modeled by Canonical Correlation Analysis. Right: Latent space modeled by our Canonical Local Preserving Latent Variable Model.

and poor generalization, but we can see that our CLPLVM model achieved better performance with limited training data.

Performance evaluation under different dimensionality reduction techniques and two latent regression methods could be seen in Table 3. Based on the Table, we made the following 5 observations: (1) inference in the latent space was nearly always better than in the original space (irrespective of the form of dimensionality reduction); (2) LPP outperformed PCA in terms of ability to preserve the manifold structure; (3) canonical methods, like CCA, KCCA, LPCCA and CLPLVM, attained better performance, owing to consistency of dimensionality reduction for both image features and 3D poses; (4) CLPLVM obtained an improvement over LPCCA; and (5) GMR outperformed linear regression with multiple predictions. We believed that (5) was due to the ability of GMR to generatively model the full density over the latent features and poses (as opposed to other more direct regression methods).

However, we could see that the performance of all algorithms in the CMU Motion Capture Database was worse than on Poser Database. In real video sequences, it is more difficult to predict a 3D human poses due to the clutter present in the background, shadows, and variations in lighting. As we used the latent variable method, our result was less sensitive to noise in the real video, which made it more accurate than other methods. Hence, we also showed subjective results of different approaches in Fig. 4, including: non-parametric regression model (kernel regression (KR)) and parametric regression models (linear regression (LR), mixture of linear regressors (MLR), mixture of experts (MoE), and Latent Mixture Gaussian Regression (LMGR) [19]). Kernel regression tended to work poorly in these cases as the data were sparse in high dimension space. The performance of mixture models degraded as the data points started to fall close to the boundary between the two experts (since we are using expectation for inference). For this reason, sometimes the performance of mixture models was

Table 2
Performance evaluation under different number of training data. For CLPLVM + GMR both expectation/mean (Exp) and the best-out-of-*K* modes, with *K* = 8, performance (B8) is reported.

Error (cm)		LR	PCA	LPP	LGMR [19]	CCA	KCCA	LPCCA [26]	CLPLVM	CLPLVM + GMR	
Train	Frame									Exp	B8
01	400	15.70	13.62	13.33	13.25	13.33	13.33	13.33	13.31	12.98	**12.94**
	800	15.72	15.65	15.48	15.31	15.47	15.48	15.48	15.47	15.09	**15.01**
01,02	1600	15.18	15.35	14.99	14.86	14.87	14.87	14.88	14.86	14.86	**14.70**
Average		15.53	14.87	14.6	14.47	14.55	14.56	14.56	14.54	14.36	**14.28**
Train time		0.02	0.48	1.81	2.06	0.31	27.30	2.55	2.23	2.47	
Test time		0.01	0.02	0.02	0.38	0.04	1.78	0.03	0.03	0.4	

Bold data give the best performance.

Table 3
Performance evaluation under different latent and regression methods.

Error (cm)		PCA		LPP		CCA		KCCA		LPCCA [26]		CLPLVM	
		4D	6D	4D	6D	4D	6D	4D	6D	4D	6D	4D	6D
LR	01	16.07	15.65	15.47	15.48	15.47	15.47	15.47	15.48	15.48	15.48	15.46	15.47
	02	15.67	15.92	14.27	14.24	14.18	14.18	14.19	14.19	14.19	14.19	14.18	14.16
	03	20.73	20.70	20.67	20.67	20.67	20.67	20.67	20.67	20.67	20.67	20.67	20.65
GMR	01	15.64	15.51	15.47	15.47	15.49	15.64	15.48	15.48	15.48	15.48	**14.93**	15.09
	02	15.62	15.24	14.01	14.30	14.18	14.12	14.19	14.19	14.15	14.17	**13.64**	13.77
	03	20.73	20.68	20.64	20.61	20.69	20.82	20.66	20.66	20.66	20.66	**20.61**	20.61

Bold data give the best performance.

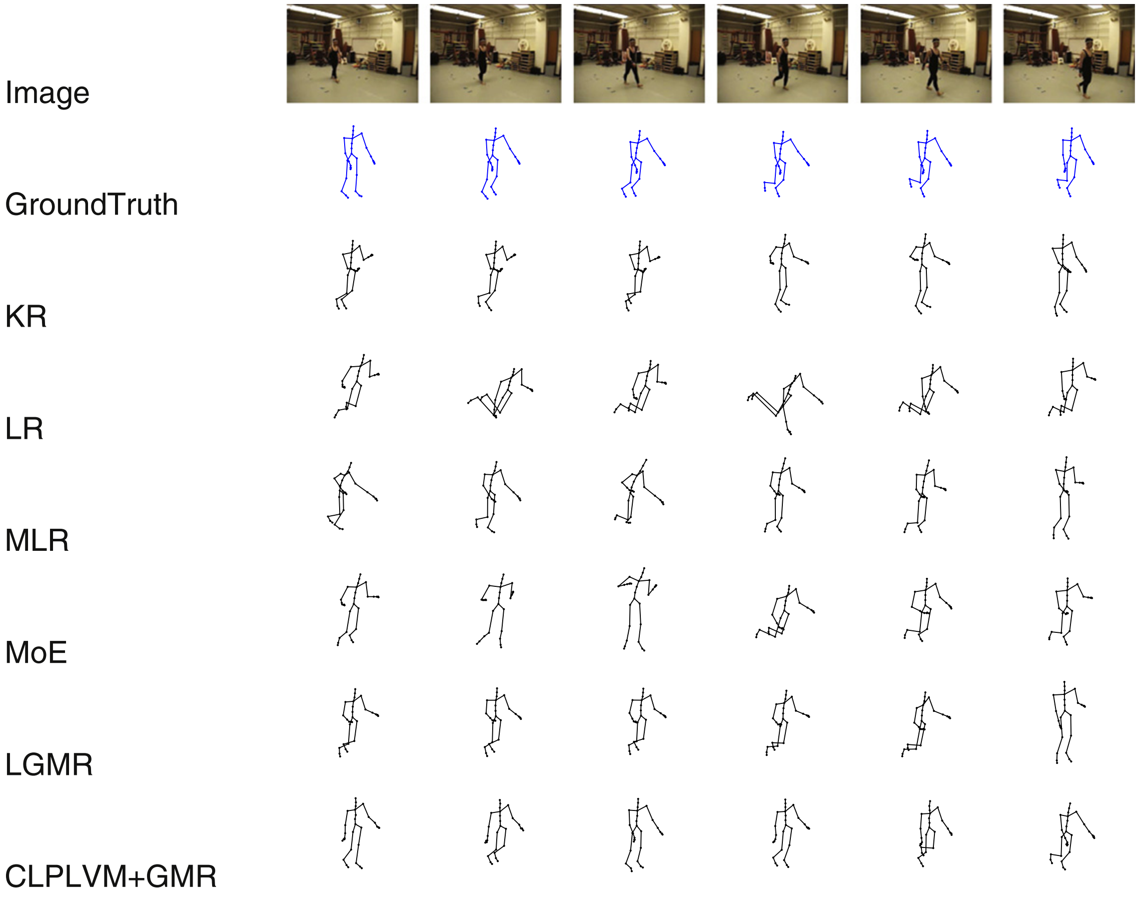

Fig. 4. Evaluation on frames 38, 48, 58, 68, 78, and 88 of sequence 04 in subject 08 from the CMU motion capture database.

lower than that of uni-modal linear regression. Our fCLPLVM + GRM model tended to produce better performance than competing methods.

5.3. Agarwal and Triggs dataset

To compare to other published techniques, we also utilized a publicly available benchmark dataset [3], that contained 1927 training and 418 test images, synthetically generated from mocap data. The pose was encoded using $M = 54$ joint angles. The image features and error metric were provided with the dataset [3]. Silhouette features were represented using 100-dimensional ($D = 100$) feature vectors encoding the image silhouette using vector-quantized shape contexts. The mean RMSE error was computed over joint angles and was measured in degrees (for details see [3]).

On this dataset we achieved an error of 6.63°, which was better than Nearest Neighbor regression, Linear Regression and Latent Gaussian Mixture Regression as reported in [17], [18] and [19] respectively. However, we could not match the performance of non-parametric shared LVMs, like Shared GPLVM and Shared KIE, that achieve errors of 6.50 and 5.95° respectively. This was not surprising given that non-parametric models could represent more complex manifold structure; however, they do come at a cost of inference and learning which, unlike in our method, is a function of the training set size.

6. Conclusions and future work

In this paper, we presented a parametric discriminative framework for 3D pose inference. Our model has a number of appealing properties, mainly it can: (1) model complex structure of the image feature and pose manifolds, (2) keep local structure in the latent space, (3) deal with multi-modalities in the data, and (4) alleviate the need for learning costly shared non-linear non-parametric manifold models. We show that our performance is comparative or superior to parametric and non-parametric models in the original high-dimensional space and in learned latent spaces. In the future, we intend to look at learning the model by combining temporal information to increase the prediction accuracy.

References

[1] H. Sidenbladh, M. Black, D. Fleet, Stochastic tracking of 3d human figures using 2d image motion, IEEE Proc. Eur. Conf. Comput. Vis. (2000) 702–718.
[2] C. Sminchisescu, B. Triggs, Covariance scaled sampling for monocular 3d body tracking, IEEE Proc. Comput. Vis. Pattern Recognit. 1 (2001) 440–447.
[3] A. Agarwal, B. Triggs, 3d human pose from silhouettes by relevance vector regression, IEEE Proc. Comput. Vis. Pattern Recognit. 2 (2004) 882–888.
[4] A. Agarwal, B. Triggs, Monocular human motion capture with a mixture of regressors, IEEE Workshop Comput. Vis. Pattern Recognit. (2005) 72–79.
[5] A. Bissacco, M. Yang, S. Soatto, Fast human pose estimation using appearance and motion via multi-dimensional boosting regression, IEEE Proc. Comput. Vis. Pattern Recognit. (2007) 1–8.
[6] L. Bo, C. Sminchisescu, Structured output-associative regression, IEEE Proc. Comput. Vis. Pattern Recognit. (2009) 2403–2410.
[7] A.M. Elgammal, C.-S. Lee, Inferring 3d body pose from silhouettes using activity manifold learning, IEEE Proc. Comput. Vis. Pattern Recognit. 2 (2004) 681–688.
[8] A. Fathi, G. Mori, Human pose estimation using motion exemplars, IEEE Proc. Inte. Conf. Comput. Vis. (2007) 1–8.
[9] F. Guo, G. Qian, Learning and inference of 3d human poses from Gaussian mixture modeled silhouettes, IEEE Proc. Int. Conf. Pattern Recognit. 2 (2006) 43–47.
[10] T. Jaeggli, E. Koller-Meier, L. Van Gool, Learning generative models for multi-activity body pose estimation, Int. J. Comput. Vis. 83 (2009) 121–134.
[11] A. Kanaujia, C. Sminchisescu, D. Metaxas, Spectral latent variable models for perceptual inference, IEEE Proc. Int. Conf. Comput. Vis. (2007) 1–8.
[12] R. Navaratnam, A. Fitzgibbon, R. Cipolla, The joint manifold model for semi-supervised multi-valued regression, IEEE Proc. Int. Conf. Comput. Vis. (2007) 1–8.

[13] G. Shakhnarovich, P. Viola, T. Darrell, Fast pose estimation with parameter-sensitive hashing, IEEE Proc. Int. Conf. Comput.Vis. (2003) 750–757.

[14] C. Sminchisescu, A. Kanaujia, Z. Li, D. Metaxas, Discriminative density propagation for 3d human motion estimation, IEEE Proc. Comput. Vis. Pattern Recognit. 1 (2005) 390–397.

[15] C. Sminchisescu, A. Kanaujia, D. Metaxas, Learning joint top-down and bottom-up processes for 3d visual inference, IEEE Proc. Comput. Vis. Pattern Recognit. 2 (2006) 1743–1752.

[16] R. Urtasun, T. Darrell, Sparse probabilistic regression for activity-independent human pose inference, IEEE Proc. Comput. Vis. Pattern Recognit. (2008) 1–8.

[17] C. Ek, P. Torr, N. Lawrence, Gaussian Process Latent Variable Models for Human Pose Estimation, In: Workshop on Machine Learning and Multimodal Interactions, 2007, pp. 132–143.

[18] L. Sigal, R. Memisevic, D.J. Fleet, Shared kernel information embedding for discriminative inference, IEEE Proc. Comput. Vis. Pattern Recognit. (2009) 2852–2859.

[19] Y. Tian, L. Sigal, H. Badino, F. De la Torre, Y. Liu, Latent Gaussian Mixture Regression for human pose estimation, In: Asian Conference on Computer Vision, 1, 2010, pp. 238–245.

[20] X. He, P. Niyogi, Locality preserving projections, Adv. Neural Inf. Process. Syst. 16 (2003) 153–160.

[21] K. Weinberger, F. Sha, L. Saul, Learning a Kernel Matrix for Nonlinear Dimensionality Reduction, In: Proceedings of the twenty-first International Conference on Machine Learning, 2004, p. 106.

[22] L. Song, A. Smola, K. Borgwardt, A. Gretton, Colored maximum variance unfolding, Adv. Neural Inf. Process. Syst. 20 (2008) 1385–1392.

[23] D. Huang, M. Storer, F. De la Torre, H. Bischof, Supervised local subspace learning for continuous head pose estimation, IEEE Conf. Comput. Vis. Pattern Recognit. (2011) 2921–2928.

[24] J. Nilsson, F. Sha, M. Jordan, Regression on manifolds using kernel dimension reduction, in: Proceedings of the 24th international conference on Machine learning, 2007, pp. 697–704.

[25] M. Kim, V. Pavlovic, Central subspace dimensionality reduction using covariance operators, IEEE Proc. Comput. Vis. Pattern Recognit. 33 (2011) 657–670.

[26] T. Sun, S. Chen, Locality preserving CCA with applications to data visualization and pose estimation, Image Vision Comput. 25 (2007) 531–543.

[27] H. Hotelling, Relations between two sets of variates, Biometrika 28 (1936) 321–377.

[28] F. De la Torre, A least-squares framework for component analysis, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 1041–1055.

[29] D. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: an overview with application to learning methods, Neural Comput. 16 (2004) 2639–2664.

[30] T. Melzer, M. Reiter, H. Bischof, Appearance models based on kernel canonical correlation analysis, Pattern Recognition, In: Special Issue on Kernel and Subspace Methods for Computer Vision, 36, 2003, pp. 1961–1971.

[31] P. Lai, C. Fyfe, Kernel and nonlinear canonical correlation analysis, Int. J. Neural Syst. 10 (2000) 365–378.

[32] B. Scholkopf, A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2002.

[33] E. Nadaraya, On estimation regression, Theor. Probab. Appl. 9 (1964) 141–142.

[34] E. Frontier, Curious Labs Poser, Comput. Softw. (2007).

[35] CMU Motion Capture Database, http://mocap.cs.cmu.edu/2004.

[36] F. Grassia, Practical parameterization of rotations using the exponential map, J. Graph. Tool. 3 (1998).

[37] A. Agarwal, B. Triggs, Recovering 3d human pose from monocular images, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 44–58.