Feature-based Pose Estimation

Cristian Sminchisescu[1,2], Liefeng Bo[3], Catalin Ionescu[4], Atul Kanaujia[5]

Abstract In this chapter we review challenges and methodology for feature-based predictive tri-dimensional human pose reconstruction, based on image and video data. We argue that reliable 3d human pose prediction can be achieved through an alliance between image descriptors that encode multiple levels of selectivity and invariance and models that are capable to represent multiple structured solutions. For monocular systems, key to reliability is the capacity to leverage prior knowledge in order to to bias solutions not only to kinematically feasible sets, but also towards typical configurations that humans are likely to assume in everyday surroundings. In this context, we discuss several predictive methods including large-scale mixture of experts, supervised spectral latent variable models and structural support vector machines, asses the impact of the various choices of image descriptors, review open problems, and give pointers to datasets and code available online.

1 Introduction

In this paper, we focus on the tri-dimensional reconstruction of human poses, represented as vectors of joint angles or 3d joint positions, based on information extracted using video cameras. We will primarily be interested in methods that are applicable to *monocular images*, although the methodology generalizes to multicamera settings. Inference from monocular images has practical applicability and cognitive relevance. In many practical settings, only a single image sequence is available as in

[1] Institute for Numerical Simulation (INS), Faculty of Mathematics and Natural Science, University of Bonn, e-mail: cristian.sminchisescu@ins.uni-bonn.de
[2] Institute for Mathematics of the Romanian Academy (IMAR), e-mail: www.imar.ro/clvp
[3] University of Washington, e-mail: lfb@cs.washington.edu
[4] INS, University of Bonn, e-mail: catalin.ionescu@ins.uni-bonn.de
[5] ObjectVideo, e-mail: atul.kanaujia@objectvideo.com

Chapter appearing in *Visual Analysis of Humans: Looking at People (T. Moeslund, A. Hilton, V. Krger, L. Sigal, Eds.), Springer, ISBN 978-0-85729-996-3, 2011.*

the case of archival movie footage, or when consumer devices are used as interface tools for gesture or activity recognition. Even when several cameras are available, fully exploiting the information in multiple views to limit uncertainty may not lead to intrinsically simpler optimization problems, as the subjects of interest may be occluded by other people or by scene elements. A robust human motion analysis system needs to handle incomplete, ambiguous or noisy observation sources that are typical of many real scenes.

From a cognitive viewpoint, *paradoxical monocular stereoscopy*[32] is the apparently seamless human ability to reconstruct the 3D structure of a scene qualitatively, using only one eye or given only a single photograph of a visual scene. As this type of monocular inference is geometrically under constrained, the mechanism that makes it possible has to do with the ability to use strong priors that link familiar scene structures and their image projection statistics. In the long run, a well-trained computer vision system should be able to leverage such priors, in order to come close to the performance of the human visual system.

The inference of human or animal motion based on images has been already studied extensively. On one hand, there exist commercial motion capture systems that represent the standard for the special effects industry, virtual and augmented reality, or medical applications and video games. These systems are very accurate but they need several calibrated and synchronized cameras, controlled illumination, and special clothing with passive markers for simplifying the image correspondence problem. On the other hand, and it is the path we pursue, there exist approaches that work with increasingly more natural images, obtained with uncalibrated, unsynchronized cameras, in natural uninstrumented environments, and filming subjects wearing their own clothing and no markers.

General Difficulties: Reconstructing the tri-dimensional human pose and motion of people at the office, on the street, or outdoors based on images acquired with a single (or even multiple) video camera(s) is one of the open problems in computer vision. The difficulties compound: people have many degrees of freedom, deform and articulate, and their appearance spans a significant range due to different body proportions and clothing. When analyzing people in realistic imaging conditions, the backgrounds cannot be controlled. Recently there is a trend to move towards operation in complex environments where people are viewed against more complex backgrounds and in a more diverse set of poses. Handling such environments would be in principle possible by means of integrated systems that combine person detection (or localization) and 3d reconstruction[55, 61, 4, 25, 13]. However, each subproblem remains difficult to solve: human detectors alone[22] cannot always handle general human poses, and even if they did, the bounding box of a person viewed against a non-uniform background still leaves a complex figure-ground search space to explore before the 3d pose can be predicted reliably (it is generally well understood that predictors based on silhouettes generalize relatively well if the input quality is good and the predictor was trained using a distribution sufficiently well sampled from the set of human poses typical of the problem domain[3, 62]). Another approach would be to use more sophisticated 2d human models for localization, with parts that mirror the true human body limbs[23, 44, 4, 46, 21]. A difficulty to overcome is the localization of people under strong perspective effects and the relatively high false postive rates. Overall, approaches based on integrated human detection and pose reconstruction remain promising, and it remains to be seen how the modeling and inference components will evolve.

Monocular Ambiguities: A major difficulty for 3D human pose inference from monocular images is the quasi-unobservability of kinematic degrees of freedom that generate motion in depth. For unknown limb (link) lengths this leads to continuous nonrigid affine folding ambiguities, but once lengths are known these reduce to twofold forwards/backwards flipping ambiguities for each link. Under no additional pose constraints, the full model thus has $\mathcal{O}(2^{\#links})$ formal inverse kinematics solutions. Reconstructing articulated 3d pose from monocular model-image point correspondences in an unconstrained human pose class is well understood to be ambiguous, according to the geometric [39, 41, 65] and computational studies[18, 50, 15, 63, 64, 45, 62] of the problem.

For real image features, the problem is more difficult to analyze geometrically (see computational studies in [18, 63, 49, 66, 45, 62]), but poses that correspond to reflective placements of limbs w.r.t. the camera rays of sight often produce only marginally different projections, with comparable likelihoods, even under similarity measures based on quite elaborate image features [19, 51, 63]. Subtle differences indeed exist between configurations that are ‘close-in-depth’ and those that are ‘far-in-depth’, but these usually give the perceived ground truth pose a relative margin only for very accurate subject models and for image observations collected under no data association errors (it remains unclear under what circumstances this is possible). In principle, shadows offer supplementary cues[5], but the key relevant regions remain difficult to identify in scenes with complex lighting and for unknown people wearing deformable clothing, and different solutions become practically indistinguishable for objects placed further away in depth from the camera.

In a sufficiently general model complexity class, monocular ambiguities can persist in video, when dynamic constraints are considered [58], and can also persist for models biased using prior knowledge [57, 45, 62] (for illustrations of both static and dynamic ambiguities, see videos at *http://sminchisescu.ins.uni-bonn.de/talks/*).

In the long run, a combination of low-complexity models and appropriate context management may produce solutions–effectively ‘controlled hallucinations’–that are unambiguous *in their class* rather than in general. This may turn out to be more effective than a ‘hardliner’ approach where models capable of representing all kinematically possible human poses are made unambiguous by fusing all ‘cues’. In fact, the hypothesis that stable visual perceptions can be formed despite extensive sets of ambiguities (*e.g.*bas-relief, pictorial depth, structure-from-motion) is not foreign to researchers in both computer vision and psychophysics [33, 32, 6].

Generative and Discriminative Methods: In this chapter we will primarily describe 3D human pose reconstruction methods based on discriminative, feed forward models[45, 48, 3, 62], which can be trained to predict pose distributions given image descriptor inputs. This strategy contrasts with the one used in generative algorithms[18, 65, 54] that search the pose space for configurations with good image likelihood (alignment). Each class of methods provides complementary advantages. Generative models are flexible at representing large classes of poses and are useful for training and hypothesis verification, but inference is expensive and good observation models are difficult to construct without simplifying assumptions. Discriminative predictors offer the promise of speed, automation and complete flexibility in selecting the image descriptor (overcomplete bases or overlapping features of the observation can be designed without simplifying independence assumptions), but have to model multivalued image-to-3D relations and their reliance on a training set makes generalization to very different poses, body proportions, or scenes where people are filmed against background clutter, problematic. (N.B. Clearly, these remain hard problems for any method, be it generative or discriminative.)

Degree of Training Data Realism: The design of multi-valued pose predictors and the temporal density propagation in conditional chains is, at present, well understood[45, 60, 62, 53], but the trade offs inherent in the acquisition of a sufficiently representative training set or the design of image descriptors with good resistance to clutter and intra-class variations was explored less. The construction of realistic *pose labeled* human databases (images of humans and their 3D poses) is inherently difficult because no existing system can provide accurate 3D ground truth for humans in real-world, non-instrumented scenes. Current solutions rely either on motion acquisition systems like Vicon, but these operate in engineered environments, where subjects wear special costumes and markers and the background is simplified, or on quasi-synthetic databases, generated by CG characters, animated using motion capture, and placed on real image backgrounds [2, 61]. In both cases, there is a risk that models learned using these training sets may not generalize well when confronted with the diversity of real world scenes. In the long run, the development of more sophisticate rendering and mixed reality frameworks, or the design of sophisticated capture sensors is likely to improve the quality of existing training sets significantly (some progress has been reported in designing systems along these lines[73]).

1.1 The Need for Structure Modeling

We will formulate human pose reconstruction as a structured learning problem, where we predict a multivariate output from a multivariate input and a joint training set (this methodology will apply to any continuous structured prediction problem in computer vision). Here, the input is the image or its descriptor (e.g. a bag of words histogram model that quantizes the occurrence of a feature over the image, for instance a local edge distribution) and the output is a scene representation, an object shape or a three-dimensional human pose. Both the inputs and the outputs are high-dimensional and strongly correlated. At basic level, image features are spatially coherent (nearby pixels more often have similar color, contrast or edge orientation than not), whereas outputs are structured due to physical constraints in the world.

For example three dimensional human poses are constrained at scene level by the ground plane and the location of typical objects, at physical level by gravitation, equilibrium and joint/body limits, and at functional level by the strong body part correlations in motions like walking, running, jumping that have regular or periodic structure, hence, at least locally, low intrinsic dimensionality (see fig. 1). Given the recent availability of training data[1, 52], there has been increasing interest in example-intensive, discriminative approaches to 3d human pose reconstruction, either based on nearest-neighbor schemes[48, 43], parametric predictors[45, 61, 3], trained using images of people and their corresponding 3d ground truth pose. A shortcoming of existing methods is their inability to model interdependencies between outputs.

Structured data can be modeled either by including sophisticated constraints into regression methods (linear or non-linear manifold assumptions [17, 61, 31, 16]), or by designing new cost functions. There is a choice of modeling correlations as part of the loss function, or as a form of regularization. One possibility is to endow manifolds with probabilistic formulations that allow mapping between data and intrinsic spaces, or computing probabilities for new datapoints [57, 36]. Additionally, graph-based geometric constraints inspired by spectral non-linear embeddings have also been integrated in a latent variable model in an *unsupervised* setting initially by [57], more recently by [29, 40] and subsequently, in a GPLVM formulation[70]. Latent variable models of this type have become popular in vision [36, 57, 71, 70, 75, 40, 29, 4, 56] predominantly as unsupervised intermediate representations, separately linked with images and used for visual inference in conjunction with particle filters[57, 71, 40] or image-based manifold predictors [29]. This turned out to be effective but is potentially suboptimal: the manifold discovered using unsupervised learning is not necessarily ideal for prediction or inference. For instance, the variance of the two distributions is by no means calibrated: the noise model of the image-to-manifold predictor could be way different than the input variance of the manifold-to-output model, negatively impacting the output estimate. Addressing such consistent end-to-end model training considerations, we will explore structured prediction methods based on manifold formulations in Sec. 3.

Another approach to structured prediction relies on max-margin formulations in conjunction with kernels defined over multivariate input and output spaces. Structural support vector machines, initially introduced for discrete state spaces [69], can be generalized to continuous outputs by learning a scoring function so that the pair corresponding to the given input-output training example ranks higher than a pair formed by the given input and any other output [76, 38]. The methods we will present can be viewed as a generalization of [69, 76] to integrated localization and state estimation problems with continuous structured inputs and outputs. A methodology adapted for simultaneous person localization and continuous pose estimation will be described in Sec. 4.

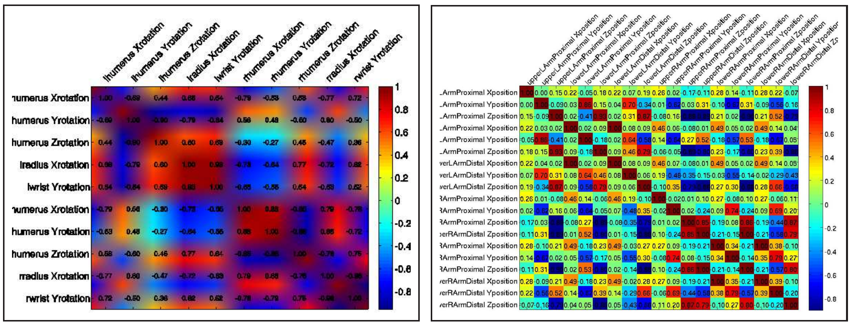

Fig. 1 Joint angle correlation statistics from human motion capture data. Left plot shows the correlations present in walking motions, whereas the right plot shows correlations for more diverse motions, including humans walking, jogging, involved in conversations or boxing. Notice not only the strong correlations in the single activity regime, but also their persistence in the more diverse motion set.

1.2 The Need for Selectivity and Invariance in Image Descriptors

A difficulty in the creation of reliable feature-based pose prediction systems is the design of image descriptors that are distinctive enough to differentiate among different poses, yet invariant to *within the same pose class differences*—people in similar stances, but differently proportioned, or photographed on different backgrounds.

Exiting methods have successfully demonstrated that bag of features or regular-grid based representations of local descriptors (*e.g.*bag of shape context features, block of SIFT features [2, 61]) can be effective at predicting 3D human poses, but the representations tend to be too inflexible for reconstruction in general scenes. It is more appropriate to view them as two useful extremes of a multilevel, hierarchical representation of images – a family of descriptors that progressively relaxes block-wise, rigid local spatial image encodings to increasingly weaker spatial models of position / geometry accumulated over increasingly larger image regions. Selecting the most competitive representation for an application – a typical set of people, motions, backgrounds or scales – reduces to either directly or implicitly learning a metric in the space of image descriptors, so that both good invariance and distinctiveness is achieved, *e.g.*, for 3D reconstruction, suppress noise by maximizing correlation within the desired pose invariance class, yet keep different classes separated, and turn off components that are close to being statistical random for the task of prediction, disregarding the class.

Multilevel, hierarchical encodings are necessary in order to obtain image descriptors with good resistance to deformations, clutter or misalignments in the dataset. But they do not entirely eliminate the need for problem dependent similarity measures for descriptor comparison, as their components may still be perturbed by clutter or may not be relevant for the task. We will thus favor hierarchical, coarse to fine image descriptors that combine multilevel image encodings and metric learning algorithms based on Canonical Correlation Analysis (CCA) and Relevant Component

Analysis (RCA). These refine and further align the image descriptors within individual pose invariance classes in order to better tolerate deformation, misalignment and clutter in the training and test sets. Below, we will review several descriptors that have been shown to be effective for human pose prediction (see [61, 28] for details):

Multilevel Spatial Blocks (MSB) is an encoding derived by Kanaujia *et al.*[28] and consists of a set of layers, each a regular grid of overlapping image blocks, with increasingly large (SIFT) descriptor cell size. Descriptors within each layer and across layers are concatenated, orderly, in order to obtain encodings of an entire image or sub-window. This is a multiscale generalization of the encoding proposed by Sminchisescu *et al.*[61] for pose prediction in the context of learning joint generative-discriminative methods.

HMAX [47] is a hierarchical, multilayer model inspired by the anatomy of the visual cortex. It alternates layers of template matching (simple cell) and max pooling (complex cell) operations in order to build representations that are increasingly invariant to scale and translation. Simple layers use convolution with local filters (template matching against a set of prototypes), in order to compute higher-order (hyper)features, whereas complex layers pool their afferent units over limited ranges, using a MAX operation, in order to increase invariance. Rather than learning the bottom layer, the model uses a bank of Gabor filter simple cells, computed at multiple positions, orientations and scales. Higher layers use simple cell prototypes, obtained by randomly sampling descriptors in the equivalent layer of a training set (k-means clustering can also be used), hence the construction of the hierarchical model has to be done stage-wise, bottom-up, as layers become available. Hyperfeatures [3] is a hierarchical, multilevel, multi-scale encoding similar in organization with HMAX, but more homogeneous in the way it repeatedly accumulates / averages template matches to prototypes (local histograms) across layers, instead of winner-takes-all MAX operations followed by template matching to prototypes.

Spatial Pyramid [37] is a hierarchical model based on encodings of spatially localized histograms, over increasingly large image regions. The bottom layer contains the finest grid, with higher layers containing coarser grids with bag of feature (SIFT) encodings computed within each one. Originally, the descriptor was used to build a pyramid kernel as a linear combination of layered, histogram intersections kernels, but it can also be used stand-alone, in conjunction with linear predictors. It aligns well with the design of our 3D predictors, that can be either linear or kernel-based.

Vocabulary Tree [42] builds a coarse-to-fine, multilevel encoding using hierarchical k-means clustering. The model is learned divisively – the training set is clustered at top level, then recursively split, with a constant branching factor, and retrained within each subgroup. Nistér & Stévenius collect measurements on a sparse grid (given by MSER interest points) and encode any path to a leaf by a single integer. This is compact and gives good results for object retrieval, but is usually not sufficiently smooth for continuous pose prediction problem, where it collapses qualitatively different poses to identical encodings. To adapt for the task, Kanaujia *et* *al.*[28] learn the same vocabulary tree, but construct stage-wise encodings by concatenating all levels. At each level we store the continuous distances to prototypes and recursively descend in the closest sub-tree. Entries in unvisited sub-trees are set to zero. For each image, the tree-based encodings of patches are accumulated on a regular grid and normalized.

The quantitative performance of different image features and metric learning methods for a monocular human pose estimation task, in conjunction with mixture of expert predictors (described in detailed in Sec. 2), is illustrated in fig. 2.

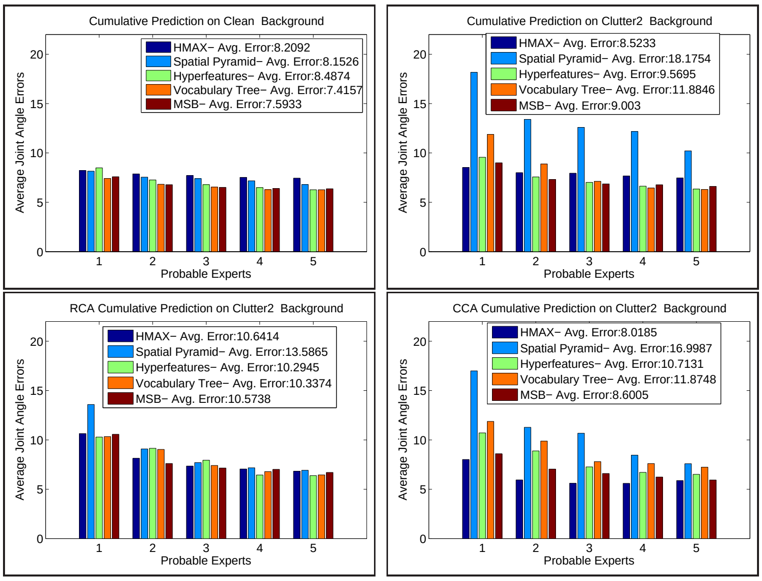

Fig. 2 Quantitative prediction errors cumulated for 5 different motions and 5 image encodings: HMAX, Hyperfeatures, Spatial Pyramid, Vocabulary Tree, Multilevel Spatial Blocks (MSB), and 2 metric learning and correlation analysis methods (RCA, CCA). A single activity-independent model (a conditional Bayesian mixture of experts) was trained on the entire dataset. Each plot shows the error in the best-k experts, for $k = 1\ldots5$, the total number of experts used. The k-th bar was computed by selecting the value closest to ground truth among the ones predicted by the most probable k experts.

2 Modeling Complex Image to Pose Relations

Monocular ambiguities and the invariance demands on the image descriptors make the modeling of image to pose relations difficult. One possibility is to train a standard function approximator (e.g. a kernel regressor) to map from image feature inputs to the continuous 3d human pose variable[3]. However, in ambiguous input regions, the accuracy of the regression model will be low. Empirical studies show that for sufficiently diverse datasets, the relations are strongly multivalued and require explicit forms of modeling the different solutions. One very powerful approach is the conditional mixture of experts models (cMoE)[60, 62].[1] The versatility of cMoE relies on a balanced combination of several attractive properties: (*i*) *conditioning on input* eliminates the need for simplifying naive Bayes assumptions, common with generative models, and allows for diverse, potentially non-independent feature functions of the input (in this case, the image) to be encoded in its descriptor. This makes possible to model non-trivial image correlations and enhances the predictive power of the input representation. (*ii*) *multivaluedness of outputs* allows for multiple plausible hypotheses – as opposed to a single one – to be faithfully represented; (*iii*) *contextual predictions* offer versatility by means of ranking (gating) functions that are paired with the experts, and adaptively score their competence in providing solutions, for each input. This allows for nuanced, finely tuned responses; (*iv*) *probabilistic consistency* enforces data modeling according to its density via formal, conditional likelihood parameter training procedures; (*v*) *Bayesian formulations and automatic relevance determination mechanisms* favor sparse models with good generalization capabilities. All these features make the cMoE model suitable for fast, automatic feedforward 3D prediction, either as a stand alone, indexing system, or as an initialization method, in conjunction with complementary visual search and feedback mechanisms [61, 55].

A significant downside of existing mixture of experts algorithms [27, 20, 7, 28, 62] is their scalability. The algorithms require an expensive double loop algorithm (an iteration within another iteration) based on Newton optimization, to compute the gate functions, a factor that makes models with more than 10,000 datapoints and large input dimension impractical to train. In this section we review computationally efficient cMoE algorithms that combine forward feature selection based on marginal likelihood and functional gradient boosting with techniques from bound optimization, in order to train models that are one order of magnitude larger (100,000 examples and up), in time that is more than one order of magnitude faster than previous methods.

Conditional Mixture of Experts: We work with a probabilistic conditional model:

$$P(x|\mathbf{r}) = \sum_{j=1}^{M} g_j(\mathbf{r}) p_j(x) \tag{1}$$

with:

$$g_j(\mathbf{r}) \equiv g(\mathbf{r}|\lambda_j) = \frac{e^{\lambda_j^\top \mathbf{r}}}{\sum_k e^{\lambda_k^\top \mathbf{r}}} \tag{2}$$

$$p_j(x) \equiv p_j(x|\mathbf{r}, \mathbf{w}_j, \sigma_j^2) = N(x|\mathbf{w}_j^\top \mathbf{r}, \sigma_j^2 \mathbf{I}) \tag{3}$$

[1] Standard kernel regression models can be viewed as a special case of a conditional mixture with a single expert ($M = 1$). We will thus not include separate derivations for kernel regression, the restriction to $M = 1$ in (1) being straightforward.

where $\mathbf{r}$ are predictor variables, x are outputs or responses, g are *input dependent* positive gates. g are normalized to sum to 1 for consistency, by construction, for any given input $\mathbf{r}$. In the model, p are Gaussian distributions (3) with variance $\sigma^2\mathbf{I}$, centered at linear regression predictions given by models with weights $\mathbf{w}$. Whenever possible, we drop the index of the experts *(but not the one of the gates)*. The weights of experts have Gaussian priors, controlled by hyperparameters α:

$$p(\mathbf{w}|\alpha) = (2\pi)^{-D/2} \prod_{d=1}^{D} \alpha_d^{1/2} \exp\{-\frac{\alpha_d w_d^2}{2}\} \tag{4}$$

with $\dim(\mathbf{w}) = D$. The parameters of the model, including experts and gates are collectively stored in $\theta = \{(\mathbf{w}_i, \alpha_i, \sigma_i, \lambda_i) \,|\, i = 1 \ldots M\}$. To learn the model, we design iterative, approximate Bayesian EM algorithms. In the E-step we estimate the posterior:

$$h_j \equiv h_j(x, \mathbf{r}|\mathbf{w}_j, \sigma_j, \lambda_j) = \frac{g_j(\mathbf{r}) p_j(x)}{\sum_{k=1}^{M} g_k(\mathbf{r}) p_k(x)} \tag{5}$$

and let $h_j^{(i)} = h_j(x^{(i)}, \mathbf{r}^{(i)})$ be the probability that the expert j has generated datapoint i. Parenthesized superscripts index datapoints. In the M-step we solve two optimization problems, one for each expert and another for the corresponding gate. The first learns the expert parameters, based on training data weighted according to the current membership estimates h. The second optimization trains the gates g to predict h. The complete log-likelihood (Q-function) for the conditional mixture of Bayesian experts can be derived as [27]:

$$Q = \sum_{i=1}^{N} \log P(x^{(i)}|\mathbf{r}^{(i)}) = \sum_{i=1}^{N} \sum_{j=1}^{M} h_j^{(i)} (\log g_j^{(i)} + \log p_j^{(i)}) = L_g + L_p \tag{6}$$

The likelihood decomposes into two factors, one for the gates and the other for the experts. The experts can be fitted independently using sparse Bayesian learning, under the change of variables $\mathbf{r}^{(t)} \leftarrow \sqrt{h^{(t)}}\mathbf{r}^{(t)}$ and $x^{(t)} \leftarrow \sqrt{h^{(t)}}x^{(t)}$. The equations for the gates are coupled and require iteration *during each* M-step. Although the problem is convex, it is computationally expensive to solve because the cost is not quadratic and the inputs are high-dimensional. A classical iteratively reweighted least squares (IRLS), or a naive Newton implementation, requires $\mathscr{O}(N(MD)^2 + (MD)^3)$ computation, multiple times during each step which is prohibitive for large problems (*e.g.*for 15 experts and 10,000 training samples with 1,000 input dimension, the computational cost becomes untenable even on today's most powerful desktops). Note that the cost of computing the Hessian (the first complexity term above) becomes higher than the one of inverting it (the second term) when the number of training samples is very large.

Training the Experts: For Bayesian learning with Gaussian priors and observation likelihoods, the expert posterior and predictive uncertainty (marked with '*') are computable in closed form:

$$\mu = \sigma^2 \Sigma \mathbf{R}\mathbf{X}, \Sigma = (\sigma^{-2}\mathbf{R}\mathbf{R}^\top + \mathbf{A})^{-1}, x^* = \mu^\top \mathbf{r}, (\sigma^2)^* = \mathbf{r}^\top \Sigma \mathbf{r} \tag{7}$$

where $\mathbf{A} = \mathrm{diag}[\alpha_1, \dots, \alpha_D]$, $\mathbf{R}$ stores the training set inputs columnwise and $\mathbf{X}$ their corresponding vector of x-outputs (see fig. 3 for illustration). The marginal likelihood of the experts is:

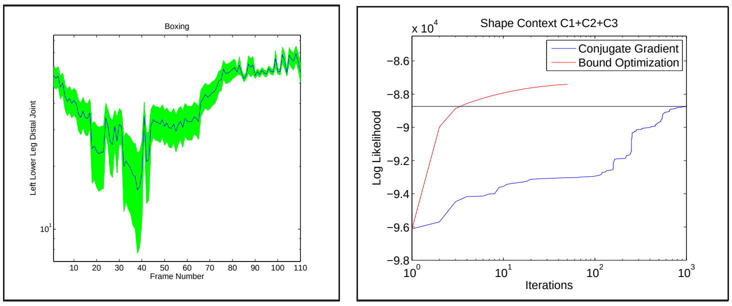

Fig. 3 *(Left)* Mean prediction and errorbars for one variable of our Bayesian model, *c.f.* (7). *(Right)* Comparative convergence behavior of our Bound Optimization (BO) and the Conjugate Gradient (CG) method when fitting the gates on a training set of 35,000 datapoints. Notice the rapid convergence of BO and that after significantly more iterations CG has not yet converged to the maximum of the log-likelihood.

$$L_p(\alpha) = \sum_{i=1}^{N} \log p(x^{(i)}|\mathbf{r}^{(i)}, \alpha, \sigma^2) = \sum_{i=1}^{N} \log \int p(x^{(i)}|\mathbf{r}^{(i)}, \mathbf{w}, \sigma^2) p(\mathbf{w}|\alpha) \mathbf{dw} = \tag{8}$$

$$= -\frac{1}{2}\{N \log 2\pi + \log|\mathbf{K}| + \mathbf{X}^\top \mathbf{K}^{-1}\mathbf{X}\} \tag{9}$$

where $\mathbf{K} = \sigma^2 \mathbf{I} + \mathbf{R}^\top \mathbf{A}^{-1}\mathbf{R}$. It can be shown that the marginal likelihood decomposes as [68]:

$$L_p(\alpha) = L_p(\alpha_{\setminus i}) + l(\alpha_i) \tag{10}$$

with

$$l(\alpha_i) = \frac{1}{2}\{\log \alpha_i - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\} \tag{11}$$

where $s_i = \mathbf{C}_i^\top \mathbf{K}_{\setminus i}^{-1} \mathbf{C}_i$ and $q_i = \mathbf{C}_i^\top \mathbf{K}_{\setminus i}^{-1} \mathbf{X}$, $\mathbf{C}_i$ collects the ith column from the matrix $\mathbf{R}^\top$, $\mathbf{K}_{\setminus i}, \alpha_{\setminus i}$ are the matrix and vector obtained with the corresponding entry of the input vector removed, and $L_p(\alpha_{\setminus i})$ is the log-likelihood.

Training the Gates: The log-likelihood component that corresponds to the gates decomposes as (λ is the $D \times M$-dimensional vector of all gate parameters λ_i):

$$L_g(\lambda) = \sum_{i=1}^{N} \sum_{j=1}^{M} h_j^{(i)} \log g_j^{(i)} = \sum_{i=1}^{N} \sum_{j=1}^{M} \{h_j^{(i)} \lambda_j^\top \mathbf{r}_i - \log \sum_{j=1}^{M} \exp(\lambda_j^\top \mathbf{r}_i)\} \tag{12}$$

For efficiency, we use bound optimization [35, 34] and maximize a surrogate function $\mathscr{F}$ with $\lambda^{(t+1)} \leftarrow \arg\max_{\lambda} \mathscr{F}(\lambda|\lambda^{(t)})$ (the upper parameter superscript indexes the iteration number in this case). This is guaranteed to monotonically increase the objective, provided that $L_g(\lambda) - \mathscr{F}(\lambda|\lambda^{(t)})$ reaches its minimum at $\lambda = \lambda^{(t)}$. A natural surrogate is the second-order Taylor expansion of the objective around $\lambda^{(t)}$, with a bound $\mathbf{H}_b$ on its second derivative (Hessian) matrix $\mathbf{H}$, so that $\mathbf{H}(\lambda) \succeq \mathbf{H}_b, \forall\lambda$:

$$\mathscr{F}(\lambda|\lambda^{(t)}) = \frac{1}{2}\lambda^\top \mathbf{H}_b \lambda + \lambda^\top(\mathbf{g}(\lambda^{(t)}) - \mathbf{H}_b\lambda^{(t)}) \tag{13}$$

The gradient and Hessian of L_g can be computed analytically:

$$\mathbf{g}(\lambda) = \sum_{i=1}^{N}(\mathbf{U}_i - \mathbf{v}_i(\lambda)) \otimes \mathbf{r}_i \tag{14}$$

with $\mathbf{U}_i = [h_1^{(i)}, \ldots, h_M^{(i)}]^\top$, $\otimes$ the Kronecker product, and $\mathbf{v}_i(\lambda) = [g_1(\mathbf{r}_i), \ldots, g_M(\mathbf{r}_i)]^\top$. The Hessian of L_g is:

$$\mathbf{H}(\lambda) = -\sum_{i=1}^{N}(\mathbf{V}_i(\lambda) - \mathbf{v}_i(\lambda)\mathbf{v}_i(\lambda)^\top) \otimes (\mathbf{r}_i\mathbf{r}_i^\top) \tag{15}$$

where $\mathbf{V}_i(\lambda) = \text{diag}[g_1(\mathbf{r}_i), \ldots, g_M(\mathbf{r}_i)]$ (the dimensionality of the Hessian is $D \times M$). The Hessian is lower bounded by a negative definite matrix which depends on the input, but *remarkably*, is independent of λ [12]:

$$\mathbf{H}(\lambda) \succeq \mathbf{H}_b \equiv -\frac{1}{2}[\mathbf{I} - \frac{11^\top}{M}] \otimes \sum_{i=1}^{N} \mathbf{r}_i\mathbf{r}_i^\top \tag{16}$$

where $1 = [1, 1, \ldots, 1]^\top$. The parameter update is based on the standard Newton step:

$$\lambda^{(t+1)} \leftarrow \lambda^{(t)} - \mathbf{H}_b^{-1}\mathbf{g}(\lambda^{(t)}) \tag{17}$$

To fit the gates we use a forward greedy algorithm that combines gradient boosting and bound optimization. It selects the variables according to functional gradient boosting [24] and optimizes the resulting sub-problems using bound optimization, as described above. To compute the functional gradient, we rewrite the objective in terms of functions $F_j(\mathbf{r}^{(i)})$. This method is applicable to any differentiable log-likelihood:

$$L_g = \sum_{i=1}^{N}\sum_{j=1}^{M}\{h_j^{(i)}F_j(\mathbf{r}^{(i)}) - \log\sum_{j=1}^{M}\exp(F_j(\mathbf{r}^{(i)})\} \tag{18}$$

The functional gradient corresponding to one component of F_j is:

$$d_j^{(i)} = \frac{\partial L_g(F_j(\mathbf{r}^{(i)}))}{\partial F_j(\mathbf{r}^{(i)})} = h_j^{(i)} - \frac{\exp(F_j(\mathbf{r}^{(i)}))}{\sum_{j=1}^{M} \exp(F_j(\mathbf{r}^{(i)}))} \tag{19}$$

with the full gradient of the jth gate assembled as $\nabla \mathbf{f}_j = [d_j^{(1)}, \ldots, d_j^{(N)}]^\top$ – the steepest descent direction in function space. For feature selection, we choose the row vector $\mathbf{v}$ of $\mathbf{R}$ with weight index not already in the active set S, and most correlated (collinear) with the gradient [24]:

$$i = \underset{k \notin S, j=1\ldots M}{\arg\max} \; |\mathbf{v}_k^\top \nabla \mathbf{f}_j| \tag{20}$$

We initialize $\lambda = \mathbf{0}$ and select the ith variable, incrementally, based on the gate parameter estimates at the previous round of selection. Once the ith variable is selected, we optimize (12) with respect to all pre-selected i variables using bound optimization. We use the solution of the previous iteration to quick-start the current optimization problem (this is convex but a good initialization saves iterations). The advantage of bound optimization in a greedy forward selection context is that we can efficiently update the Hessian bound using the Woodbury inversion identity. Thus, the cost of each iteration is $\mathcal{O}(cNMD)$ where c is a small constant, and the total cost of selecting the k variables is $\mathcal{O}(kNMD)$. When the specified number of variables is reached, we terminate. Unlike gradient boosting where the only current selected variable is optimized, we also perform back-fitting [72], *i.e.*optimize all selected variables in each round. To speed-up computation, it is possible to optimize the weights of the gating networks sequentially–fix the weights of other gating networks than the one currently optimized–the problem in (19). This requires the solution to a sequence of k-dimensional problems (usually $k << D$) and can be significantly cheaper than updating all gate parameters simultaneously, especially when denser (less sparse) models are desired. To sparsify the gating network, one can consider forward selection ideas based on maximizing the marginal likelihood, along the same lines as used for experts. However, the computational cost of this approach is high even for fast Bayesian approximations to multinomial classification. Differently from Bayesian regression, there is no analytical expression for the marginal likelihood, hence we have to resort on Laplace approximation. But this only works around the maximized posterior point, so we have to recompute the most probable weight and the corresponding Hessian matrix after adding or deleting an input entry (or basis function). For large problems this operation is computationally prohibitive.

3 Manifolds: Supervised Spectral Latent Variable Models

A variety of computer vision and machine learning tasks require the analysis of high-dimensional ambient signals, *e.g.*2d images, 3d range scans or data obtained from human motion capture systems. The goal is to learn compact, perceptual (latent) models of the data generation process and use them to interpret new measurements. For example, the variability in an image sequence filming a rotating teapot is non-linearly produced by latent factors like rotation variables and the lighting direction. Our subjective, *perceived* dimensionality partly mirrors the latent factors, being significantly smaller than the one directly *measured* – the high-dimensional sequence of image pixel vectors. Similarly, filming a human running or walking requires megabytes of wildly varying images, yet in a representation that properly correlates the human joint angles, the intrinsic dimensionality is effectively 1 – the phase of the walking cycle. The argument can go on, but underlines the intuitive idea that the space of all images is much larger than the set of physically possible ones, which, in turn is larger than the one typically observed in most every day's scenes. If this is true, perceptual inference cannot proceed without an appropriate, arguably probabilistic model of correlation, a natural way to link perceptual and measured inferences. This implies a non-linear subset, or a manifold assumption, at least in the large-sample regime: the low-dimensional perceptual structure lives in the high-dimensional space of direct observations. To unfold it, we need faithful, topographic representations of the observed data – effectively forms of continuity and locality: nearby observations should map to nearby percepts and faraway observations should map faraway. Given this, we want to be able to consistently answer the following questions: How to represent a percept or an image? What is the probability of an observed image? What is the probability of a given percept? What is the conditional probability of a percept given an image and vice-versa?

In this section we introduce probabilistic models with geometric properties in order to marry spectral embeddings, parametric latent variable models, and structured image predictors. We refer to these conditional probabilistic constructs, defined on top of an irregular grid (or unfolding) obtained from a spectral embedding as Supervised Spectral Latent Variable Models (SSLVM).

We will work with a training set $(\mathbf{r}_i, \mathbf{y}_i), i = 1 \ldots N$ with inputs $\mathbf{r}$ and outputs $\mathbf{y}$, both multivariate. We construct a latent variable model with intermediate (hidden) representation $\mathbf{x}$ with distribution that preserves geometric constraints among outputs $\mathbf{y}$, and at the same time offers good predictive power when regressed against the inputs $\mathbf{r}$.

3.1 Conditional Latent Variable Model

The joint distribution over latent and output variables, conditioned on inputs is:

$$p(\mathbf{y}, \mathbf{x}|\mathbf{r}, \theta, \delta) = p(\mathbf{y}|\mathbf{x}, \theta) p(\mathbf{x}|\mathbf{r}, \delta) \tag{21}$$

with (θ, δ) parameters of the two distributions (in the sequel dropped whenever not essential for readability). The conditional response is calculated by integrating the latent space:

$$p(\mathbf{y}|\mathbf{r}) = \int p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}|\mathbf{r}) \mathbf{dx} \approx \frac{1}{S} \sum_{s=1}^{S} p(\mathbf{y}|\mathbf{x}^{(s)}) \tag{22}$$

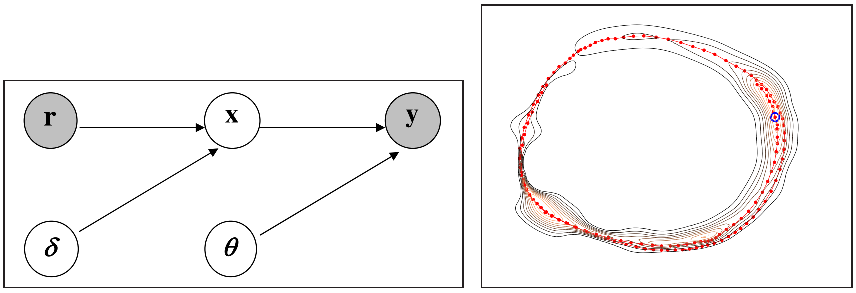

Fig. 4 *(Left)* Graphical Model of SSLVM. Shaded nodes indicate observed random variables ($\mathbf{y}$ being observed only in training). We jointly learn two conditional distributions $p(\mathbf{x}|\mathbf{r}) = p(\mathbf{x}|\mathbf{r},\delta)$ and $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x},\theta)$ with a constraint that the geometry of the outputs, as encoded in distances between datapoints $d(\mathbf{y}_i,\mathbf{y}_j)$ is implicitly preserved among their corresponding latent pre-image expectations $d(\mathrm{E}(\mathbf{x}|\mathbf{y}_i),\mathrm{E}(\mathbf{x}|\mathbf{y}_j))$. *(Right)* shows the conditional latent space distribution $p(\mathbf{x}|\mathbf{y})$ for a walking model given only the left arm variables are observed (shoulder and elbow, here 5 out of 41 variables); the latent coordinate corresponding to the complete vector of 'ground truth' joint angles is shown with a circle. Notice that 3 modes that arise due to uncertainty from missing data.

Since we work with non-linear conditional models $p(\mathbf{x}|\mathbf{r})$ and $p(\mathbf{y}|\mathbf{x})$ the integral in (22) cannot be computed analytically. Hence, we approximate using a Monte Carlo estimate based on S samples drawn from the conditional $p(\mathbf{x}|\mathbf{r})$ [67].[2] This is tractable and efficient because the latent conditional is usually low-dimensional and has, for our choice of models, a convenient parametric form—either Gaussian for regression or Gaussian mixture in the case of conditional mixtures of experts models. Specifically, we use:

$$p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x},\theta = (\mathbf{W},\Sigma)) = N(\mathbf{W}\Phi(\mathbf{x}),\Sigma) \tag{23}$$

and $p(\mathbf{x}|\mathbf{r})$ given by a mixture of expert model described in Sec 2. The latent space conditional is obtained using Bayes' rule:

$$p(\mathbf{x}|\mathbf{y},\mathbf{r}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\mathbf{r})}{p(\mathbf{y}|\mathbf{r})} = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\mathbf{r})}{\frac{1}{S}\sum_{s=1}^{S} p(\mathbf{y}|\mathbf{x}^{(s)})} \tag{24}$$

For pairs of training data i and MC latent samples s, we abbreviate $p_{(s,i)} = p(\mathbf{x}^{(s)}|\mathbf{y}_i,\mathbf{r}_i)$. Notice how the choice of latent conditional $p(\mathbf{x}|\mathbf{r})$ influences the membership probabilities in (24). We can compute either the conditional mean or the mode (better for multimodal distributions) in latent space, using the same MC integration method used for (22):

$$\mathrm{E}\{\mathbf{x}|\mathbf{y}_n,\mathbf{r}_n\} = \int p(\mathbf{x}|\mathbf{y}_n,\mathbf{r}_n)\mathbf{x}d\mathbf{x} = \sum_{s=1}^{S} p_{(s,n)}\mathbf{x}_s \tag{25}$$

[2] Sampled configurations have parenthesized superscripts; subscripts index training datapoints.

$$s_{max} = \arg\max_{s} p_{(s,n)} \tag{26}$$

The model has the components for consistent calculations in both the latent and output spaces: $p(\mathbf{x}|\mathbf{r})$ gives the latent space distribution, (22) the output marginal, (23) provides the conditional (or mapping) from latent to output, and (25) and (26) give the mean or mode of the mapping from output to latent space. More accurate but also more expensive mode-finding approximations can be obtained by direct gradient ascent on (24). Latent conditionals given partially observed $\mathbf{y}$ vectors are easy to compute, using (24). The distribution on $\mathbf{y}$ is Gaussian and unobserved components can be integrated analytically – this effectively removes them from the mean and the corresponding lines and columns of the covariance. Computations like these are useful as often outputs can have missing entries, *e.g.*marker drop-outs in a motion capture system during training, 'pattern completion' or restoration of an image under the latent model during testing, see *e.g.* fig. 4, right.

Implicit Latent Geometric Constraints: Assume that distances between outputs $\mathbf{y}$ are stored in a vector $\mathbf{D}$ of size N^2, with entries $d(\mathbf{y}_i, \mathbf{y}_j)$ with d an arbitrary similarity function that can be the Euclidean distance, a Gaussian centered at the first argument, or a geodesic distance in the data graph (these will be used to model distance preserving constraints like the ones encountered in PCA/MDS, Laplacian Eigenmaps or ISOMAP, respectively). Consider a similar vector $\mathbf{L}$ of corresponding latent space distances $d(\mathrm{E}(\mathbf{x}|\mathbf{y}_i), \mathrm{E}(\mathbf{x}|\mathbf{y}_j))$, where $\mathrm{E}(\mathbf{x}|\mathbf{y})$ is the conditional expectation of latent variable $\mathbf{x}$ given $\mathbf{y}$, $c.f.$ (25). We use vectors of pairwise distances among outputs and their corresponding latent conditional expectations in order to construct an implicit geometric constraint (or penalty) in latent space:

$$C = (\mathbf{D} - \mathbf{L})(\mathbf{D} - \mathbf{L})^{\top} \tag{27}$$

which is 0 if output distances are preserved in latent space and large otherwise. Notice that $d(\mathbf{x}_i, \mathbf{x}_j)$ gives the distance between the i-th and j-th point in data or latent space, rather than the relative spatial positions of points in data and latent space. Clearly $d(\mathbf{x}, \mathbf{x}) = 0$. However, our method does not explicitly require distance properties. We can work, in principle, with any similarity measure such as the inner product. Notice also the dependence of the penalty on $\mathrm{E}(\mathbf{x}|\mathbf{y}) = \mathrm{E}(\mathbf{x}|\mathbf{y}, \theta)$, which is a function of the current parameters θ of the latent to output model $p_\theta(\mathbf{y}|\mathbf{x})$, $c.f.$ (23), (24) and (25). The penalty ensures that under the latent space posterior, the implicit latent space distances $\mathrm{E}(\mathbf{x}|\mathbf{y}_i)$ among configurations $\mathbf{x}$ that correspond to datapoints $\mathbf{y}_i$ under the current model θ, are preserved (distances $d(\mathbf{y}_i, \mathbf{y}_j)$). Notice that no matter the spectral constraint used, the resulting model is highly non-linear: latent variables depend non-linearly on inputs and and outputs depend non-linearly on latent the variables.

The *implicit geometric constraint regularizer* in (27) requires the calculation of conditional expectations for the latent variables given output data, which may at first appear more complex. Notice that the regularizer does not depend explicitly on latent variables, hence we do not optimize latent variables explicitly – this would be underconstrained, prone to overfitting, and computationally prohibitive for models with more than a few latent dimensions. Furthermore, the proposed expression can be approximated efficiently by considering the inner product as a form distance function:

$$d(\mathrm{E}(\mathbf{x}|\mathbf{y}_i), \mathrm{E}(\mathbf{x}|\mathbf{y}_j)) = \sum_{s=1}^{S} \sum_{t=1}^{S} p_{(s,i)} p_{(t,j)} \mathbf{x}_s^\top \mathbf{x}_t \tag{28}$$

Typically, most $p_{(s,i)}$ will be close to zero. Removing those does not reduce the evaluation of distance functions but offers large speedups when computing its derivative, since $\mathbf{x}_s^\top \mathbf{x}_t$ can be stored ahead of time.

Learning Algorithm: We learn the conditional model in (21) by optimizing a penalized likelihood criterion that consists of the marginal likelihood (22) averaged over a dataset and the geometric penalty on the latent space (27):

$$\mathscr{L}(\theta, \delta) = \log \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{r}_i) - \lambda C = \sum_{i=1}^{N} \log p(\mathbf{y}_i|\mathbf{r}_i) - \lambda C = \tag{29}$$

$$= \sum_{i=1}^{N} \log \sum_{s=1}^{S} p(\mathbf{y}_i|\mathbf{x}^{(s)}, \mathbf{r}_i) - \lambda C \tag{30}$$

and λ is the regularizing (trade-off) parameter. In practice we will optimize a cost $\mathscr{F}$ that is the penalized expectation of the complete data log-likelihood $\mathscr{L}_c$:

$$\mathscr{F} = < \mathscr{L}_c(\theta, \delta) > -\lambda C = \sum_{i=1}^{N} \sum_{s=1}^{S} p_{(s,i)} \log p(\mathbf{y}_i|\mathbf{x}^{(s)}) - \lambda C \tag{31}$$

We train the model by estimating $p(\mathbf{x}|\mathbf{r})$ and $p(\mathbf{y}|\mathbf{x})$ in alternation. We initialize the latent coordinates $\mathbf{x}_i$ corresponding to the given output data $\mathbf{y}_i$ using dimensionality reduction, based on the type of geometric constraint we wish to impose, *e.g.*PCA, ISOMAP or Laplacian Eigenmaps, and we use their corresponding distances between datapoints $d(\mathbf{y}_i, \mathbf{y}_j)$ in the penalty term C, see (27). Then, we first train $p(\mathbf{x}|\mathbf{r})$ based on $(\mathbf{r}_i, \mathbf{x}_i)$ data, and $p(\mathbf{y}, \mathbf{x})$ by sampling from the learned model $p(\mathbf{x}|\mathbf{r})$ with target data $\mathbf{y}_i$ and the constraint C. Then, we alternate between generating data $(\mathrm{E}(\mathbf{x}|\mathbf{y}_i), \mathbf{r}_i)$ for training the input model $p(\mathbf{x}|\mathbf{r})$, and training the output model $p(\mathbf{y}|\mathbf{x})$ using EM: in the *E-step* we estimate the membership probabilities *c.f*. (24), and in the *M-step* we solve a penalized weighted regression problem as in (31), with weights given by (24) and penalty given by C (27). Notice that C changes since $\mathrm{E}(\mathbf{x}|\mathbf{y})$ is a function of the current $\theta = (\mathbf{W}, \Sigma)$ parameters of $p(\mathbf{y}|\mathbf{x})$, *c.f*. (25) and (23).

4 Structural SVM: Joint Localization and State Estimation

In the previous sections we have described predictive methods that are accurate, scalable, and can handle the ambiguity in the image to pose relationships by explicitly representing multiple solutions using mixture of experts models. We have also shown how this methodology can be constrained and made even faster by imposing low-dimensional manifold constraints. One of the remaining problems is to integrate person detection and pose estimation. One option is to combine the mixture of experts prediction with a front-end person detector or localizer. This approach has been pursued in [61] and [55], respectively.

In this section, we consider an alternative approach where we learn a structured scoring function in the joint space of image bounding box coordinates of a person and his or her continuous 3d pose. We consider a learning setting, where we are given a set of the input-output pairs $\{\mathbf{r}_i, \mathbf{z}_i\}_{i=1}^N$, with N is the size of training set and $\mathbf{z} \in Z$ are interdependent outputs. We aim to learn a function that best represents the relationship between inputs and outputs. In structured learning, the discriminative function is a linear combination of joint features

$$g(\mathbf{r}) = \text{argmin}_{\mathbf{z}} f_{\mathbf{w}}(\mathbf{r},\mathbf{z}) = \mathbf{w}^\top \Psi(\mathbf{r},\mathbf{z}) \tag{32}$$

where $\mathbf{w}$ is a parameter vector and $\Psi(\mathbf{r},\mathbf{z})$ is a feature vector induced by a joint kernel $K(\mathbf{r},\mathbf{z},\mathbf{r}',\mathbf{z}') = \Psi(\mathbf{r},\mathbf{z})^\top \Psi(\mathbf{r}',\mathbf{z}')$. The specific form of joint features $\Psi(\mathbf{r},\mathbf{z})$ is problem-dependent, an aspect that will be discussed in in the sequel. The scoring function $f_{\mathbf{w}}(\mathbf{r},\mathbf{z})$ can be interpreted as a compatibility that measures how well the output $\mathbf{z}$ matches the input $\mathbf{r}$.

To learn the discriminative function, $f_{\mathbf{w}}(\mathbf{r},\mathbf{z})$, the structural SVM (structSVM) maximizes the generalized maximum margin loss:

$$\begin{aligned}
&\min_{\mathbf{w},\xi} \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C\xi \qquad (33)\\
&s.t. \quad \forall(\bar{\mathbf{z}}_1,\ldots,\bar{\mathbf{z}}_N) \in Z^N\\
&\frac{1}{N}\mathbf{w}^\top \sum_{j=1}^N [\Psi(\mathbf{r}_j,\mathbf{z}_j) - \Psi(\mathbf{r}_j,\bar{\mathbf{z}}_j)] \geq \frac{1}{N}\sum_{j=1}^N \Delta(\mathbf{z}_j,\bar{\mathbf{z}}_j) - \xi
\end{aligned}$$

where $\Delta(\mathbf{z}_j,\bar{\mathbf{z}}_j)$ is a loss function that should decrease as the output $\bar{\mathbf{z}}_j$ approaches the ground truth $\mathbf{z}_j$. We use the so-called '*1*-slack formulation', which is equivalent to the '*n*-slack' analogue, but is more efficient in conjunction with cutting plane algorithms (we use) due to a significantly smaller dual problem [26].

Under infinitely many constraints, standard duality does not apply. However, for any small δ we can assume a finite δ-cover of our data domain, where the constraints are locally uniform. This allows to recast the problem into one with finite (yet large) number of constraints. In this case, the primal/dual theory implies that the parameter $\mathbf{w}$ has the form:

$$\mathbf{w} = \frac{1}{N}\sum_{\bar{\mathbf{Z}} \in Z^N} \alpha_{\bar{\mathbf{Z}}} \sum_{j=1}^N [\Phi(\mathbf{r}_j,\mathbf{z}_j) - \Phi(\mathbf{r}_j,\bar{\mathbf{z}}_j)] \tag{34}$$

where $\bar{\mathbf{Z}} = (\bar{\mathbf{z}}_1,\ldots,\bar{\mathbf{z}}_\mathbf{N})$.

Joint Kernel for Location and State Estimation: For joint localization and state (pose) estimation, the input is an image, $\mathbf{r}$, and the output is the bounding box of the object *together* with its corresponding continuous state (*e.g.*2d or 3d pose):

$\mathbf{z} = (\mathbf{y}, \mathbf{x})$. We use $\mathbf{r}|\mathbf{y}$ to denote the feature vector of image regions restricted within the bounding box instantiated by $\mathbf{y}$. Here, we consider a joint kernel where the combined feature vector can be written as a tensor product over the two corresponding subspaces

$$\Psi(\mathbf{r}, \mathbf{z}) = \phi(\mathbf{r}|\mathbf{y}) \otimes \varphi(\mathbf{x}) \tag{35}$$

where $\otimes$ denotes the tensor product, $\phi(\mathbf{r}|\mathbf{y})$ is the feature induced by the kernel $K_{r|y}(\mathbf{r}|\mathbf{y}, \mathbf{r}'|\mathbf{y}')$ defined over the image region and $\varphi(\mathbf{x})$ is the feature vector induced by the state/pose kernel $K_x(\mathbf{x}, \mathbf{x}')$. For the tensor product feature vector, the joint location and state kernel is chosen to have the following form [69]:

$$K(\mathbf{r}, \mathbf{z}, \mathbf{r}', \mathbf{z}') = \Psi(\mathbf{r}, \mathbf{z})^{\top} \Psi(\mathbf{r}', \mathbf{z}') = K_{r|y}(\mathbf{r}|\mathbf{y}, \mathbf{r}'|\mathbf{y}') K_x(\mathbf{x}, \mathbf{x}') \tag{36}$$

Eq. (36) implies that the joint kernel is a product of components computed over image regions within the bounding box and the corresponding state, respectively. This tensor product feature is rather general and can handle many types of structured outputs, including multiclass and sequential constraints. In vision, kernels are used to compare statistics or image features, *e.g.*as inner products of histograms defined over regions. This includes for example, bag-of-feature models, regular grids like HOG, as well as spatial pyramids based on weighted combinations of histogram intersection kernels computed at multiple levels of image encoding. An attractive feature of histogram kernels is that partially overlapping image regions share underlying statistics.

On the other hand, since we work with continuous states, $\varphi(\mathbf{x})$ is a feature vector induced by the kernels defined over continuous variables. Concurrently, we wish the kernel function to be normalized to prevent the slack variable ξ from diverging to infinity for some outputs. One possible (but by no means the only) choice satisfying these desiderata is the Gaussian kernel. This is defined over continuous variables and its 2-norm $\|\varphi(\mathbf{x})\|_2 = \sqrt{K_x(\mathbf{x}, \mathbf{x})} = 1$. Thus, for most of our experiments, the state/pose kernel has the form: $K_x(\mathbf{x}, \mathbf{x}') = \exp(-\gamma_x \|\mathbf{x} - \mathbf{x}'\|^2)$.

When designing a similarity measure between two different input-output features, intuitively, we wish: 1) input-output pairs with distant inputs should be dissimilar; and 2) input-output pairs whose inputs are nearby but outputs are distant should also be dissimilar; otherwise stated *only* the input-output pairs with both similar inputs *and* similar outputs should be similar. The joint kernel we use satisfies the above conditions because it is the product of two kernels defined over inputs and outputs, respectively; hence its value is small if any one of the two kernel component values is small (dissimilar). In this respect, the joint kernel is more expressive than a classical kernel, only defined over inputs, where the input-output pairs with similar inputs but dissimilar outputs have negative impact and can pull the estimate in contradictory directions. For localization and state estimation, the advantage of a joint kernel over separable ones is that given a test image, the training data with dissimilar states/poses from the test input will have reduced impact on the bounding box search and estimate. This may explain why continuous structSVM achieves better performance than support vector regression (SVR) and other unstructured methods in our experience[25]. In addition, the model also includes search/ranking in the space of possible object locations in the image, described next.

Output Loss Function: The output loss $\Delta(\mathbf{z},\mathbf{z}')$ should reflect how well $\mathbf{z}$ approaches the ground truth output $\mathbf{z}'$. Within our joint tensor product kernel formulation, the loss function definition should be compatible with both the image and the state/pose kernels. For the image kernel, we adapt the score used in the PASCAL visual object challenge [8]:

$$\Delta_y(\mathbf{y},\mathbf{y}') = 1 - \frac{\text{Area}(\mathbf{y} \cap \mathbf{y}')}{\text{Area}(\mathbf{y} \cup \mathbf{y}')} \tag{37}$$

where the quality of object localization is based on the amount of area overlap, where $\text{Area}(\mathbf{y} \cap \mathbf{y}')$ is the intersection of the two bounding boxes $\mathbf{y}$ and $\mathbf{y}'$, and $\text{Area}(\mathbf{y} \cup \mathbf{y}')$ is their union. For state/pose estimation, it is natural to consider the loss function as a square distance in the reproducing kernel Hilbert space induced by the kernel function $K_x(\mathbf{x},\mathbf{x}')$

$$\Delta_x(\mathbf{x},\mathbf{x}') = \|\varphi(\mathbf{x}) - \varphi(\mathbf{x}')\|^2 = K_x(\mathbf{x},\mathbf{x}) + K_x(\mathbf{x}',\mathbf{x}') - 2K_x(\mathbf{x},\mathbf{x}') \tag{38}$$

This implies that if a state is far from the ground truth, it has high loss, otherwise small loss. We define the joint output loss as the weighted sum of losses for object localization and state estimation

$$\Delta(\mathbf{z},\mathbf{z}') = \gamma\Delta_y(\mathbf{y},\mathbf{y}') + (1-\gamma)\Delta_x(\mathbf{x},\mathbf{x}') \tag{39}$$

with $0 \leq \gamma \leq 1$ balancing the two terms.

Cutting Plane Algorithm: When training the model with continuous outputs, the number of constraints is infinite, and it is infeasible to solve the optimization (33) for all the constraints. Fortunately, the maximum margin loss has a sparsity-promoting effect, with most constraints inactive in the final solution. The cutting plane algorithm creates a nested sequence of successively tighter relaxations of the original optimization problem and finds a small set of active constraints that ensures a sufficiently accurate solution—a practical training method (see our fig. 5a). The algorithm starts with an empty working set $S = \varnothing$ of constraints. At each iteration, it finds the most violated constraint for the i-th training input If the amount of violation exceeds the current value of the slack variable ξ by more than ε, the potential support vector $\overline{\mathbf{Z}} = (\overline{\mathbf{z}}_1,\ldots,\overline{\mathbf{z}}_N)$ is added to the working set $S = S \cup \overline{\mathbf{Z}}$. After the working set is updated, the optimization (33) is solved in the dual with constraints $\overline{Z} \in S$ and The algorithm stops when no violation is larger than the desired precision ε. Notice that at the first iteration, the set of constraints S is empty–in this case finding the most violated constraint simplifies to maximizing the loss $\Delta(\mathbf{z}_i,\mathbf{z})$ with respect to the output $\mathbf{z}$. Unlike the n-slack formulation, the dual problem for the *1*-slack usually remains compact, as only a single constraint is added per iteration.

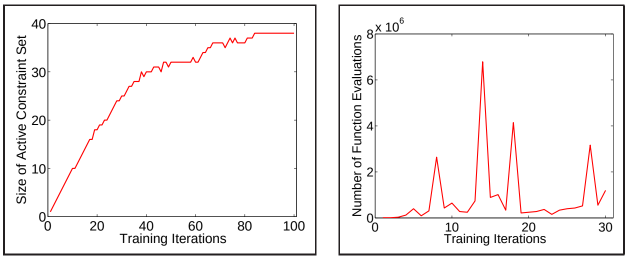

Fig. 5 *(a, Left)* Size of the active constraint set (*i.e.*the number of support vectors) as function of iteration. The number of support vectors saturates at about 80 iterations, indicating convergence of the cutting plane algorithm. Notice that support vectors (points with non-zero dual variables) in the *1*-slack formulation are linear combinations of multiple examples and no longer correspond to a single training data point. *(b, Right)* Number of function evaluations for the branch-and-bound algorithm in the training stage of a joint structural SVM, based on bag-of-words SIFT (image size 640 × 480). Notice that the number of function evaluations is significantly higher during some of the iterations compared to the others, confirming that the hardness of search is closely linked to the structural SVM parameters, **w**.

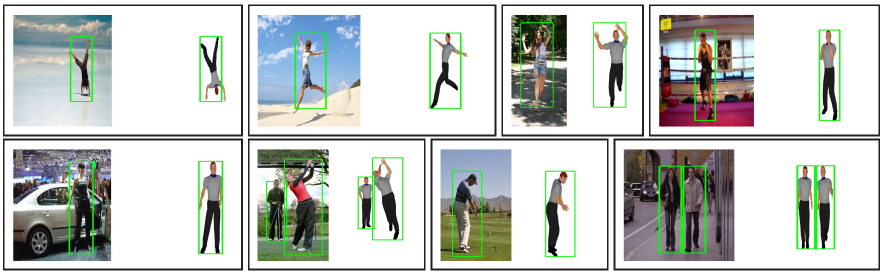

Fig. 6 Localization and 3d human pose reconstructions in real world images, for structSVM models trained on a dataset of quasi-real images. The framework allows the localization and reconstruction of multiple people.

4.1 Code and Datasets

Datasets to evaluate the performance of 3d human pose reconstruction algorithms are still relatively few and the breadth of motions and data capturing contexts is still limited. One dataset that contains a variety of motions (sports, conversations, *etc*.) is available from CMU[1], but this does not have accompanying image data synchronized with the motion capture data. The dataset is useful for constructing human pose priors, but cannot be used to evaluate the performance of 3d human pose reconstruction algorithms. The HumanEva dataset from Brown[52] is probably one of the most comprehensive benchmarks for 3d human pose reconstruction available at the moment. It contains a variety of motions (walking, running, jogging, boxing) captured from several subjects. Image data captured from several synchronized, calibrated video streams is also available for training and performance evaluation.

The dataset and baseline particle filtering code for human pose estimation can be found on HumanEva's website[52], *http://vision.cs.brown.edu/humaneva/*. Discriminative pose prediction code, including multivalued predictors based on mixture of experts[11], structured predictors based on twin Gaussian Processes[10], and spectral latent variable models[29, 9] can be found on our website, at: *http://sminchisescu.ins.uni-bonn.de/code*.

5 Challenges and Open Problems

Over the past 15 years, there has been significant progress in 3d human pose recovery, both conceptually—understanding the limitations of existing likelihood or search algorithms, and deriving improved procedures and conceptually new models and image features to overcome them—and practically, by building systems that can reconstruct shape and perform fine body measurements from images [53], model complex physical properties[14, 74], recover 3d human motion from multiple views [18, 30] or reconstruct human pose from complex monocular video footage like dancing or movies[65, 28].

For real-world applications, one of the main challenges is to automatically understand people in-vivo, and in an integrated fashion. Typical computations are to find the people, infer their poses, recognize their activities, the objects and the interactions.

Many of the existing human motion analysis systems are still complex to construct, computationally expensive, and cannot seamlessly deal with significant structural variability, multiple interacting people and severe occlusion or lighting changes. A convincing transition between the laboratory and the real world remains to be realized. A shortcoming of existing systems is their inability to provide a satisfactory solution to the problem of model selection and the management of the level of detail. Anyone who watches movies, TV or other sources of video can easily notice to what limited extent the people are visible in close-up full-body views. The subjects are frequently occluded by other people, by scene elements, or simply clipped to obtain partial views for artistic or practical reasons–the camera operator tends to focus on the relevant aspects of the gesture and pose rather than the entire body, and so should, perhaps, the computer vision algorithms. This suggests that monolithic models may have to be replaced with a set of models that can flexibly represent partial views and multiple level of detail. Level of detail, albeit considered in the different sense of 2D, vs 2.5D vs. 3D may need to be modeled and inferred automatically. The extent of 2D to 3D lifting, ranging from none, partial to full, should ideally be calibrated to the task and correlated with the degree of scene observability, or with the image resolution. Working with multiple models raises inference questions of deciding what model is appropriate, switching between different models in a tractable manner, or managing the trade-off between run-time inference and indexing based on prior knowledge. The interaction between human pose reconstruction and activity recognition will probably emerge as a natural solution within the broader scope of coherent scene understanding.

Acknowledgments: This work has been supported in part by the European Commission under MCEXT-025481 and by CNCSIS-UEFISCU under project PN II-RU-RC-2/2009. This chapter reviews research presented in [28, 11, 9, 25].

References

1. CMU Human Motion Capture DataBase. Available online at http://mocap.cs.cmu.edu/search.html, 2003.
2. A. Agarwal and B. Triggs. Hyperfeatures – Multilevel Local Coding for Visual Recognition. In *European Conference on Computer Vision*, 2006.
3. A. Agarwal and B. Triggs. A local basis representation for estimating human pose from cluttered images. In *ACCV*, 2006.
4. M. Andriluka, S. Roth, and B. Schiele. People Tracking-by-Detection and People-Detection-by-Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
5. A. Balan, M. Black, H Haussecker, and L. Sigal. Shining a Light on Human Pose: On Shadows, Shading and the Estimation of Pose and Shape. In *IEEE International Conference on Computer Vision*, 2007.
6. B. Battu, A. Krappers, and J. Koenderink. Ambiguity in Pictorial Depth. *Perception*, 36, 2007.
7. C. Bishop and M. Svensen. Bayesian mixtures of experts. In *Uncertainty in Artificial Intelligence*, 2003.
8. Matthew B. Blaschko and Christoph H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, pages 2–15, 2008.
9. L. Bo and C. Sminchisescu. Supervised Spectral Latent Variable Models. *Artificial Intelligence and Statistics*, 2009.
10. L. Bo and C. Sminchisescu. Twin Gaussian Processes for Structured Prediction. *International Journal of Computer Vision*, 2010.
11. L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. Fast Algorithms for Large Scale Conditional 3D Prediction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
12. D. Böhning. Multinomial logistic regression algorithm. *Annals of Inst. of Stat. Math.*, 44:197–200, 2001.
13. L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *IEEE International Conference on Computer Vision*, 2009.
14. M. Brubaker and D. Fleet. The Kneed Walker for Human Pose Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
15. K. Choo and D. Fleet. People Tracking Using Hybrid Monte Carlo Filtering. In *IEEE International Conference on Computer Vision*, 2001.
16. R. D. Cook. *Regression Graphics*. Wiley Inter-Science, 1988.
17. C. Cortes, M. Mohri, and J. Weston. A general regression technique for learning transductions. In *International Conference on Machine Learning*, pages 153–160, 2005.
18. J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2000.
19. J. Deutscher, A. Davidson, and I. Reid. Articulated Partitioning of High Dimensional Search Spacs associated with Articulated Body Motion Capture. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2001.
20. D. Edwards and S. Lauritzen. The TM algorithm for maximising a conditional likelihood function. *Biometrika*, 88(4):961–972, 2001.

21. M. Eichner and V. Ferrari. We are family: Joint Pose Estimation of Multiple Persons. In *European Conference on Computer Vision*, 2010.
22. P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010.
23. P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1), 2005.
24. J. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.
25. C. Ionescu, L. Bo, and C. Sminchisescu. Structural SVM for Visual Localization and Continuous State Estimation. In *IEEE International Conference on Computer Vision*, 2009.
26. T. Joachims, T. Finley, and C. Yu. Cutting-plane training of structural svms. *Machine Learning*, 2009.
27. M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, (6):181–214, 1994.
28. A. Kanaujia, C. Sminchisescu, and D. Metaxas. Semi-Supervised Hierarchical Models for 3D Human Pose Reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
29. A. Kanaujia, C. Sminchisescu, and D. Metaxas. Spectral Latent Variable Models for Perceptual Inference. In *IEEE International Conference on Computer Vision*, volume 1, 2007.
30. R. Kehl, M. Bray, and L. V. Gool. Full body tracking from multiple views using stochastic sampling. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.
31. M. Kim and V. Pavlovic. Dimensionality reduction using covariance operator inverse regression. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
32. J. Koenderink. Pictorial Relief. *Phil. Trans. R. Soc. Lond. A*, 356, 1998.
33. J. Koenderink and A. van Doorn. The Internal Representation of Solid Shape with Respect to Vision. *Biological Cybernetics*, 32(3), 1979.
34. B. Krishnapuram, L. Carin, M. T. Figueiredo, and A. J. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):957–968, 2005.
35. K. Lange, D. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *J. Computational and Graphical Statistics*, 9:1–59, 2001.
36. N. Lawrence. Probabilistic non-linear component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, (6):1783–1816, 2005.
37. S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
38. Y. LeCun, S. Chopra, R. Hadsell, R. M. Ranzato, and F. Huang. A Tutorial on Energy-Based Learning. In *Predicting Structured Data*. MIT Press, 2006.
39. H. J. Lee and Z. Chen. Determination of 3D Human Body Postures from a Single View. *Computer Vision, Graphics and Image Processing*, 30:148–168, 1985.
40. Z. Lu, M. Carreira Perpinan, and C. Sminchisescu. People Tracking with the Laplacian Eigenmaps Latent Variable Model. In *Advances in Neural Information Processing Systems*, 2007.
41. D. Morris and J. Rehg. Singularity Analysis for Articulated Object Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 289–296, 1998.
42. D. Nistér and H. Stévenius. Scalable recognition with a vocabulary tree. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
43. R. Poppe. Evaluating example-based human pose estimation: Experiments on HumanEva sets. In *HumanEva Workshop CVPR*, 2007.
44. D. Ramanan and C. Sminchisescu. Training Deformable Models for Localization. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
45. R. Rosales and S. Sclaroff. Learning Body Pose Via Specialized Maps. In *Advances in Neural Information Processing Systems*, 2002.

46. B. Sapp, A. Toshev, and B. Taskar. Cascaded Models for Articulated Pose Estimation. In *European Conference on Computer Vision*, 2010.
47. T. Serre, L Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 994–1000, Washington, DC, USA, 2005.
48. G. Shakhnarovich, P. Viola, and T. Darrell. Fast Pose Estimation with Parameter Sensitive Hashing. In *IEEE International Conference on Computer Vision*, 2003.
49. H. Sidenbladh and M. Black. Learning Image Statistics for Bayesian Tracking. In *IEEE International Conference on Computer Vision*, 2001.
50. H. Sidenbladh, M. Black, and D. Fleet. Stochastic Tracking of 3D Human Figures Using 2D Image Motion. In *European Conference on Computer Vision*, 2000.
51. H. Sidenbladh, M. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conference on Computer Vision*, 2002.
52. L. Sigal, A. Balan, and M. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 2006.
53. L. Sigal, A. Balan, and M. J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Advances in Neural Information Processing Systems*, 2007.
54. L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking Loose-limbed People. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.
55. L. Sigal and M. Black. Predicting 3d people from 2d pictures. In *Articulated Motion and Deformable Objects*, 2006.
56. L. Sigal, R. Memisevic, and D. Fleet. Shared kernel information embedding for discriminative inference. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2009.
57. C. Sminchisescu and A. Jepson. Generative Modeling for Continuous Non-Linearly Embedded Visual Inference. In *International Conference on Machine Learning*, pages 759–766, Banff, 2004.
58. C. Sminchisescu and A. Jepson. Variational Mixture Smoothing for Non-Linear Dynamical Systems. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 608–615, Washington D.C., 2004.
59. C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional Visual Tracking in Kernel Space. In *Advances in Neural Information Processing Systems*, 2005.
60. C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative Density Propagation for 3D Human Motion Estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 390–397, 2005.
61. C. Sminchisescu, A. Kanaujia, and D. Metaxas. Learning Joint Top-down and Bottom-up Processes for 3D Visual Inference. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.
62. C. Sminchisescu, A. Kanaujia, and D. Metaxas. BM^3E: Discriminative Density Propagation for Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
63. C. Sminchisescu and B. Triggs. Building Roadmaps of Local Minima of Visual Models. In *European Conference on Computer Vision*, volume 1, pages 566–582, Copenhagen, 2002.
64. C. Sminchisescu and B. Triggs. Hyperdynamics Importance Sampling. In *European Conference on Computer Vision*, volume 1, pages 769–783, Copenhagen, 2002.
65. C. Sminchisescu and B. Triggs. Kinematic Jump Processes for Monocular 3D Human Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 69–76, Madison, 2003.
66. C. Sminchisescu and B. Triggs. Mapping Minima and Transitions in Visual Models. *International Journal of Computer Vision*, 61(1), 2005.
67. C. Sminchisescu and M. Welling. Generalized Darting Monte-Carlo. In *Artificial Intelligence and Statistics*, volume 1, 2007.
68. M. Tipping and A. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In *AISTATS*, 2003.

69. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.
70. R. Urtasun, D. Fleet, A. Geiger, J. Popovic, T. Darrell, and N. Lawrence. Topologically-constrained latent variable models. In *International Conference on Machine Learning*, 2008.
71. R. Urtasun, D. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking in small training sets. In *IEEE International Conference on Computer Vision*, 2005.
72. P. Vincent and Y. Bengio. Kernel matching pursuit. *Machine Learning*, 48:165–187, 2002.
73. D. Vlasic, R. Adelsberger, G. Vannucci, J. Barwell, M. Gross, W. Matusik, and J. Popovic. Practical Motion Capture in Everyday Surroundings. In *SIGGRAPH*, 2007.
74. M. Vondrak, L. Sigal, and O. C. Jenkins. Physical Simulation for Probabilistic Motion Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.
75. J. Wang, D. J. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.
76. J. Weston, B. Schölkopf, O. Bousquet, T. Mann, and W. Noble. Joint kernel maps. In *LNCS*, 2005.