Recovering 3D Human Pose from Monocular Images

Ankur Agarwal and Bill Triggs

***Abstract*— We describe a learning based method for recovering 3D human body pose from single images and monocular image sequences. Our approach requires neither an explicit body model nor prior labelling of body parts in the image. Instead, it recovers pose by direct nonlinear regression against shape descriptor vectors extracted automatically from image silhouettes. For robustness against local silhouette segmentation errors, silhouette shape is encoded by histogram-of-shape-contexts descriptors. We evaluate several different regression methods: ridge regression, Relevance Vector Machine (RVM) regression and Support Vector Machine (SVM) regression over both linear and kernel bases. The RVMs provide much sparser regressors without compromising performance, and kernel bases give a small but worthwhile improvement in performance. Loss of depth and limb labelling information often makes the recovery of 3D pose from single silhouettes ambiguous. We propose two solutions to this: the first embeds the method in a tracking framework, using dynamics from the previous state estimate to disambiguate the pose; the second uses a mixture of regressors framework to return multiple solutions for each silhouette. We show that the resulting system tracks long sequences stably, and is also capable of accurately reconstructing 3D human pose from single images, giving multiple possible solutions in ambiguous cases. For realism and good generalization over a wide range of viewpoints, we train the regressors on images resynthesized from real human motion capture data. The method is demonstrated on a 54-parameter full body pose model, both quantitatively on independent but similar test data, and qualitatively on real image sequences. Mean angular errors of 4–5 degrees are obtained — a factor of 3 better than the current state of the art for the much simpler upper body problem.**

***Index Terms*— Computer vision, human motion estimation, machine learning, multivariate regression, Relevance Vector Machine**

I. Introduction

We consider the problem of estimating and tracking 3D configurations of complex articulated objects from monocular images, *e.g.* for applications requiring 3D human body pose and hand gesture analysis. There are two main schools of thought on this. *Model-based approaches* presuppose an explicitly known parametric body model, and estimate the pose either by directly inverting the kinematics (which has many possible solutions and requires known image positions for each body part) [28], or by numerically optimizing some form of model-image correspondence metric over the pose variables, using a forward rendering model to predict the images (which is expensive and requires a good initialization, and the problem always has many local minima [25]). An important subcase is *model-based tracking*, which focuses on tracking the pose estimate from one time step to the next starting from a known initialization, based on an approximate dynamical model [9,23]. In contrast, *learning based approaches* try to avoid the need for explicit initialization and accurate 3D modelling and rendering, and to capitalize on the fact that the set of *typical* human poses is far smaller than the set of kinematically possible ones, by estimating (learning) a model that directly recovers pose estimates from observable image quantities. In particular, *example based methods* explicitly store a set of training examples whose 3D poses are known, and estimate pose by searching for training image(s) similar to the given input image, and interpolating from their poses [5,18,22,27].

In this paper we take a learning based approach, but instead of explicitly storing and searching for similar training examples, we use sparse Bayesian nonlinear regression to distill a large training database into a single compact model that has good generalization to unseen examples. Given the high dimensionality and intrinsic ambiguity of the monocular pose estimation problem, the selection of appropriate image features and good control of overfitting is critical for success. We are not aware of previous work on pose estimation that directly addresses these issues. Our strategy is based on the sparsification and generalization properties of our nonlinear regression algorithm, which is a form of the *Relevance Vector Machine (RVM)* [29]. RVMs have been used earlier, *e.g.* to build kernel regressors for 2D displacement updates in correlation-based patch tracking [33]. Human pose recovery is significantly harder — more ill-conditioned and nonlinear and much higher dimensional — but by selecting a sufficiently rich set of image descriptors, it turns out that we can still obtain enough information for successful regression. Loss of depth and limb labelling information often makes the recovery of 3D pose from single silhouettes ambiguous. We propose two solutions to this. The first embeds the method in a tracking framework, using dynamics from the previous state estimate to disambiguate the pose. The second uses a mixture of regressors framework to return multiple possible solutions for each silhouette, allowing accurate pose reconstructions from single images. When working with a sequence of images, these solutions are fed as input to a multiple hypothesis tracker to give the most likely estimate for each time step.

Previous work: There is a good deal of prior work on human pose analysis, but relatively little on directly learning 3D pose from image measurements. Brand [8] models a dynamical manifold of human body configurations with a Hidden Markov Model and learns using entropy minimization, Athitsos and Sclaroff [4] learn a perceptron mapping between the appearance and parameter spaces, and Shakhnarovich *et al*

[22] use an interpolated-k-nearest-neighbor learning method. Human pose is hard to ground truth, so most papers in this area [4,8,18] use only heuristic visual inspection to judge their results. However Shakhnarovich *et al* [22] used a human model rendering package (POSER from Curious Labs) to synthesize ground-truthed training and test images of 13 d.o.f. upper body poses with a limited ($\pm 40^\circ$) set of random torso movements and view points. In comparison, our regression algorithm estimates full body pose and orientation (54 d.o.f.) — a problem whose high dimensionality would really stretch the capacity of an example based method such as [22]. Like [11,22], we used POSER to synthesize a large set of training and test images from different viewpoints, but rather than using random synthetic poses, we used poses taken from real human motion capture sequences. Our results thus relate to real data.

Several publications have used the image locations of the centre of each body joint as an intermediate representation, first estimating these centre locations in the image, then recovering the 3D pose from them. Howe *et al* [12] develop a Bayesian learning framework to recover 3D pose from known centres, based on a training set of pose-centre pairs obtained from resynthesized motion capture data. Mori & Malik [18] estimate the centres using shape context image matching against a set of training images with pre-labelled centres, then reconstruct 3D pose using the algorithm of [28]. These approaches show that using 2D joint centres as an intermediate representation can be an effective strategy, but we have preferred to estimate pose directly from the underlying local image descriptors as we feel that this is likely to prove both more accurate and more robust, also providing a generic framework for directly estimating and tracking any prespecified set of parameters from image observations.

As regards tracking, some approaches have learned dynamical models for specific human motions [19,20]. Particle filters and MCMC methods have been widely used in probabilistic tracking frameworks *e.g.* [23,31]. Most of these methods use an explicit generative model to compute observation likelihoods. We propose a discriminatively motivated framework in which dynamical state predictions are directly fused with descriptors computed from the observed image. Our algorithm is related to Bayesian tracking, but we eliminate the need for both an explicit body model that is projected to predict image observations, and a corresponding error model that is used to evaluate the likelihood of the observed image given this projection. A brief description of our regression based scheme is given is [1] and its first extension to resolve ambiguities using dynamics within the regression is described in [2].

Overview of the approach: We represent 3D body pose by 55-D vectors $\mathbf{x}$ including 3 joint angles for each of the 18 major body joints. This choice corresponds to the motion capture data that we use to train the system (details in section II-B). The input images are reduced to 100-D observation vectors $\mathbf{z}$ that robustly encode the shape of a human image silhouette. Given a set of labelled training examples $\{(\mathbf{z}_i, \mathbf{x}_i) \,|\, i = 1 \ldots n\}$, the RVM learns a smooth reconstruction function $\mathbf{x} = \mathbf{r}(\mathbf{z})$, valid over the region spanned by the training points. The function is a weighted linear combination $\mathbf{r}(\mathbf{z}) \equiv \sum_k \mathbf{a}_k \, \phi_k(\mathbf{z})$ of a prespecifed set of scalar basis functions $\{\phi_k(\mathbf{z}) \,|\, k = 1 \ldots p\}$. In our tracking framework, to help to disambiguate pose in cases where there are several possible reconstructions, the functional form is extended to include an approximate preliminary pose estimate $\check{\mathbf{x}}$, $\mathbf{x} = \mathbf{r}(\check{\mathbf{x}}, \mathbf{z})$. (See section V.) At each time step, a state estimate $\check{\mathbf{x}}_t$ is obtained from the previous two pose vectors using an autoregressive dynamical model, and this is used to compute the basis functions, which now take the form $\{\phi_k(\check{\mathbf{x}}, \mathbf{z}) \,|\, k = 1 \ldots p\}$. Section VI gives an alternative method for handling ambiguities by returning multiple possible 3D configurations corresponding to a silhouette. The functional form is extended to a probabilistic mixture $p(\mathbf{x}) \sim \sum_k \pi_k \delta(\mathbf{x}, \mathbf{r}_k)$ allowing each reconstruction $\mathbf{r}_k$ to output a different solution.

Our solutions are well-regularized in the sense that the weight vectors $\mathbf{a}_k$ are damped to control over-fitting, and sparse in the sense that many of them are zero. Sparsity occurs because the RVM actively selects only the 'most relevant' basis functions — the ones that really need to have nonzero coefficients to complete the regression successfully. A sparse solution obtained by the RVM allows the system to select relevant input *features* (components) in case of a linear basis ($\phi_k(\mathbf{z}) = k^{th}$ component of $\mathbf{z}$). For a kernel basis — $\phi_k(\mathbf{z}) \equiv K(\mathbf{z}, \mathbf{z}_k)$ for some kernel function $K(\mathbf{z}_i, \mathbf{z}_j)$ and centres $\mathbf{z}_k$ — relevant training examples are selected, allowing us to prune a large training dataset and retain only a minimal subset.

Organization: §II describes our image descriptors and body pose representation. §III gives an outline of our regression methods. §IV details the recovery of 3D pose from single images using this regression, discussing the RVM's feature selection properties but showing that ambiguities in estimating 3D pose from single images cause occasional 'glitches' in the results. §V describes our first solution to this problem: a tracking based regression framework capable of resolving these ambiguities, with results from our novel tracker in §V-B. §VI describes an alternative solution: a mixture of regressors based approach incorporated in a multiple hypothesis tracker. Finally, §VII concludes with some discussions and directions for future work.

II. REPRESENTING IMAGES AND BODY POSES

Directly regressing pose on input images requires a robust, compact and well-behaved representation of the observed image information and a suitable parametrization of the body poses that we wish to recover. To encode the observed images we use robust descriptors of the shape of the subject's image silhouette, and to describe our body pose, we use vectors of joint angles.

A. Images as Shape Descriptors

Silhouettes: Of the many different image descriptors that could be used for human pose estimation, and in line with [4,8], we have chosen to base our system on image silhouettes.

Silhouettes have three main advantages: *(i)* They can be extracted moderately reliably from images, at least when robust background- or motion-based segmentation is available and

Fig. 1. Different 3D poses can have very similar image observations, causing the regression from image silhouettes to 3D pose to be inherently multi-valued. The legs are the arms are reversed in the first two images, for example.

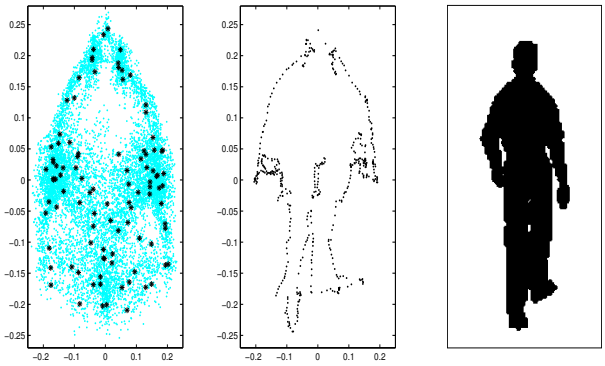


Fig. 2. (Left) The first two principal components of the distribution of all shape context vectors from a training data sequence, with the k-means centres superimposed. The average-over-human-silhouettes like form arises because (besides finer distinctions) the context vectors encode approximate spatial position on the silhouette: a context at the bottom left of the silhouette receives votes only in its upper right bins, *etc*. (Centre) The same projection for the edge-points of a single silhouette (shown on the right).

problems with shadows are avoided; *(ii)* they are insensitive to irrelevant surface attributes like clothing colour and texture; *(iii)* they encode a great deal of useful information about 3D pose without the need of any labelling information [1].

Two factors limit the performance attainable from silhouettes: *(i)* Artifacts such as shadow attachment and poor background segmentation tend to distort their local form. This often causes problems when global descriptors such as shape moments are used (as in [4,8]), as every local error pollutes each component of the descriptor: to be robust, shape descriptors need to have good *locality*. *(ii)* Silhouettes make several discrete and continuous degrees of freedom invisible or poorly visible (see fig. 1). It is difficult to tell frontal views from back ones, whether a person seen from the side is stepping with the left leg or the right one, and what are the exact poses of arms or hands that fall within (are "occluded" by) the torso's silhouette. Including interior edge information within the silhouette [22] is likely to provide a useful degree of disambiguation in such cases, but is difficult to disambiguate from, *e.g*. markings on clothing.

Shape Context Distributions: To improve resistance to segmentation errors and occlusions, we need a robust silhouette representation. The first requirement for robustness is *locality*. Histogramming edge information is a good way to encode local shape robustly [17,6], so we begin by computing local descriptors at regularly spaced points on the edge of the silhouette. We use shape contexts (histograms of local edge pixels into log-polar bins [6]) to encode silhouette shape quasi-locally over a range of scales, computing the contexts in local regions defined by diameter roughly equal to the size of a limb. In our application we assume that the vertical is preserved, so to improve discrimination, we do not normalize contexts with respect to their dominant local orientations as originally proposed in [6]. The silhouette shape is thus encoded as a distribution (in fact, as a noisy multibranched curve, but we treat it as a distribution) in the 60-D shape context space. (In our implementation, shape contexts contain 12 angular $\times$ 5 radial bins, giving rise to 60 dimensional histograms.) Matching silhouettes is therefore reduced to matching these distributions in shape context space. To implement this, a second level of histogramming is performed: we reduce the distributions of all points on each silhouette to 100-D histograms by vector quantizing the shape context space. Silhouette comparison is thus finally reduced to a comparison of 100-D histograms. The 100 centre codebook is learned once and for all by running k-means on the combined set of context vectors of all of the training silhouettes. See fig. 2. (Other centre selection methods give similar results.) For a given silhouette, a 100-D histogram $\mathbf{z}$ is built by allowing each of its context vectors to vote softly into the few centre-classes nearest to it, and accumulating scores of all context vectors. This *soft* voting reduces the effects of spatial quantization, allowing us to compare histograms using simple Euclidean distance, rather than, say, Earth Movers Distance [21]. (We have also tested the normalized cellwise distance $\|\sqrt{\mathbf{p}_1}-\sqrt{\mathbf{p}_2}\|^2$, with very similar results.) The histogram-of-shape-contexts scheme gives us a reasonable degree of robustness to occlusions and local silhouette segmentation failures, and indeed captures a significant amount of pose information (see fig. 3).

B. Body Pose as Joint Angles

We recover 3D body pose (including orientation w.r.t. the camera) as a real 55-D vector $\mathbf{x}$, including 3 joint angles for each of the 18 major body joints. The subject's overall azimuth (compass heading angle) θ can wrap around through 360°. To maintain continuity, we actually regress $(a, b) = (\cos\theta, \sin\theta)$ rather than θ, using $\mathrm{atan2}(b, a)$ to recover θ from the not-necessarily-normalized vector returned by regression. So we have $3\times18+1 = 55$ parameters.

We stress that our framework is inherently 'model-free' and is independent of the choice of this pose representation. The system itself has no explicit body model or rendering model,

[1] We do not believe that any representation (Fourier coefficients, *etc*.) based on treating the silhouette shape as a continuous parametrized curve is appropriate for this application: silhouettes frequently change topology (*e.g*. when a hand's silhouette touches the torso's one), so parametric curve-based encodings necessarily have discontinuities w.r.t. shape.

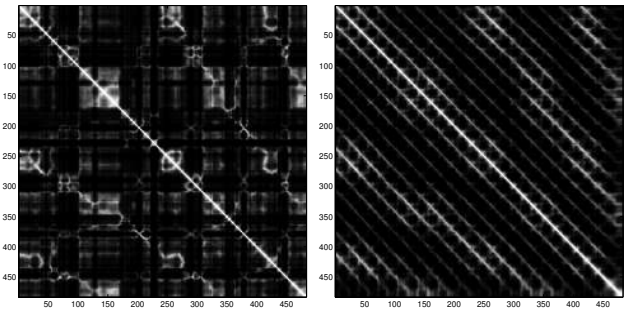


Fig. 3. Pairwise similarity matrices for (left) image silhouette descriptors and (right) true 3D poses, for a 483-frame sequence of a person walking in a decreasing spiral. The light off-diagonal bands that are visible in both matrices denote regions of comparative similarity linking corresponding poses on different cycles of the spiral. This indicates that our silhouette descriptors do indeed capture a significant amount of pose information. (The light SW-NE ripples in the 3D pose matrix just indicate that the standing-like poses at the middle of each stride have mid-range joint values, and hence are closer on average to other poses than the 'stepping' ones at the end of strides).

and no knowledge of the 'meaning' of the motion capture parameters that it is regressing — it simply learns to predict these from silhouette data. Similarly, we have not sought to learn a minimal representation of the true human pose degrees of freedom, but simply to regress the original motion capture based training format, and our regression methods handle such redundant output representations without problems.

The motion capture data was taken from the public website www.ict.usc.edu/graphics/animWeb/ humanoid. Although we use real motion capture data for joint angles, we do not have access to the corresponding image silhouettes, so we currently use a graphics package, POSER from Curious Labs, to synthesize suitable training images, and also to visualize the final reconstruction. This does unfortunately involve the use of a synthetic body model, but we stress that this model is not part of our system and would not be needed if real motion capture data with silhouettes were available.

III. Regression Methods

This section describes the regression methods that we have evaluated for recovering 3D human body pose from the above image descriptors. Here we follow standard regression notation, representing the output pose by real vectors $\mathbf{y} \in \mathbb{R}^m$ and the input shape as vectors $\mathbf{x} \in \mathbb{R}^d$. [2]

For most of the paper, we assume that the relationship between $\mathbf{x}$ and $\mathbf{y}$ — which a priori, given the ambiguities of pose recovery, might be multi-valued and hence relational rather than functional — can be approximated functionally as a linear combination as a prespecified set of basis functions:

$$\mathbf{y} \;=\; \sum_{k=1}^{p} \mathbf{a}_k \, \phi_k(\mathbf{x}) + \boldsymbol{\epsilon} \;\equiv\; \mathbf{A}\,\mathbf{f}(\mathbf{x}) + \boldsymbol{\epsilon} \qquad (1)$$

[2]However note that in subsequent sections, outputs (3D-pose vectors) will be denoted by $\mathbf{x} \in \mathbb{R}^{55}$ and inputs will be instances from either the observation space, $\mathbf{z} \in \mathbb{R}^{100}$, or the joint (predicted) state + observation space, $(\mathbf{x}^\top, \mathbf{z}^\top)^\top \in \mathbb{R}^{155}$.

Here, $\{\phi_k(\mathbf{x}) \,|\, k = 1 \ldots p\}$ are the basis functions, $\mathbf{a}_k$ are $\mathbb{R}^m$-valued weight vectors, and $\boldsymbol{\epsilon}$ is a residual error vector. For compactness, we gather the weight vectors into an $m \times p$ weight matrix $\mathbf{A} \equiv (\mathbf{a}_1 \;\; \mathbf{a}_2 \;\cdots\; \mathbf{a}_p)$ and the basis functions into a $\mathbb{R}^p$-valued function $\mathbf{f}(\mathbf{x}) = (\phi_1(\mathbf{x}) \;\; \phi_2(\mathbf{x}) \;\cdots\; \phi_p(\mathbf{x}))^\top$. To allow for a constant offset $\mathbf{A}\mathbf{f}+\mathbf{b}$, we can include $\phi(\mathbf{x}) \equiv 1$ in $\mathbf{f}$.

To train the model (estimate $\mathbf{A}$), we are given a set of training pairs $\{(\mathbf{y}_i, \mathbf{x}_i) \,|\, i = 1 \ldots n\}$. In this paper we will usually use the Euclidean norm to measure $\mathbf{y}$-space prediction errors, so the estimation problem is of the form:

$$\mathbf{A} \;:=\; \arg\min_{\mathbf{A}} \left\{ \sum_{i=1}^{n} \|\mathbf{A}\,\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i\|^2 \;+\; R(\mathbf{A}) \right\} \qquad (2)$$

where $R(-)$ is a regularizer on $\mathbf{A}$. Gathering the training points into an $m \times n$ output matrix $\mathbf{Y} \equiv (\mathbf{y}_1 \;\; \mathbf{y}_2 \;\cdots\; \mathbf{y}_n)$ and a $p \times n$ feature matrix $\mathbf{F} \equiv (\mathbf{f}(\mathbf{x}_1) \;\; \mathbf{f}(\mathbf{x}_2) \;\cdots\; \mathbf{f}(\mathbf{x}_n))$, the estimation problem takes the form:

$$\mathbf{A} \;:=\; \arg\min_{\mathbf{A}} \left\{ \; \|\mathbf{A}\,\mathbf{F} - \mathbf{Y}\|^2 + R(\mathbf{A}) \; \right\} \qquad (3)$$

Note that the dependence on $\{\phi_k(-)\}$ and $\{\mathbf{x}_i\}$ is encoded entirely in the numerical matrix $\mathbf{F}$.

A. Ridge Regression

Pose estimation is a high dimensional and intrinsically ill-conditioned problem, so simple least squares estimation — setting $R(\mathbf{A}) \equiv \mathbf{0}$ and solving for $\mathbf{A}$ in least squares — typically produces severe overfitting and hence poor generalization. To reduce this, we need to add a smoothness constraint on the learned mapping, for example by including a damping or regularization term $R(\mathbf{A})$ that penalizes large values in the coefficient matrix $\mathbf{A}$. Consider the simplest choice, $R(\mathbf{A}) \equiv \lambda \|\mathbf{A}\|^2$, where λ is a regularization parameter. This gives the *ridge* regressor, or *damped least squares* regressor, which minimizes

$$\|\mathbf{A}\,\tilde{\mathbf{F}} - \tilde{\mathbf{Y}}\|^2 \;:=\; \|\mathbf{A}\,\mathbf{F} - \mathbf{Y}\|^2 + \lambda\,\|\mathbf{A}\|^2 \qquad (4)$$

where $\tilde{\mathbf{F}} \equiv (\mathbf{F} \;\; \lambda\,\mathbf{I})$ and $\tilde{\mathbf{Y}} \equiv (\mathbf{Y} \;\; \mathbf{0})$. The solution can be obtained by solving the linear system $\mathbf{A}\,\tilde{\mathbf{F}} = \tilde{\mathbf{Y}}$ (*i.e.* $\tilde{\mathbf{F}}^\top \mathbf{A}^\top = \tilde{\mathbf{Y}}^\top$) for $\mathbf{A}$ in least squares[3], using QR decomposition or the normal equations. Ridge solutions are not equivariant under scaling of inputs, so we usually standardize the inputs (*i.e.* scale them to have unit variance) before solving.

λ must be set large enough to control ill-conditioning and overfitting, but not so large as to cause overdamping (forcing $\mathbf{A}$ towards $\mathbf{0}$ so that the regressor systematically underestimates the solution).

B. Relevance Vector Regression

Relevance Vector Machines (RVMs) [29,30] are a sparse Bayesian approach to classification and regression. They introduce Gaussian priors on each parameter or group of parameters, each prior being controlled by its own individual

[3]In case a constant offset $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$ is included, this vector $\mathbf{b}$ must not be *damped* and hence the system takes the form $(\mathbf{A} \;\; \mathbf{b})\;\tilde{\mathbf{F}} = \tilde{\mathbf{Y}}$ where $\tilde{\mathbf{F}} \equiv \begin{pmatrix} \mathbf{F} & \lambda\,\mathbf{I} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}$ and $\tilde{\mathbf{Y}} \equiv (\mathbf{Y} \;\; \mathbf{0})$.

RVM Training Algorithm

1) Initialize $\mathbf{A}$ with ridge regression. Initialize the running scale estimates $a_{\text{scale}} = \|\mathbf{a}\|$ for the components or vectors $\mathbf{a}$.
2) Approximate the $\nu \log \|\mathbf{a}\|$ penalty terms with "quadratic bridges", the gradients of which match at a_{scale}. *I.e.* the penalty terms take the form $\frac{\nu}{2}(\mathbf{a}/a_{\text{scale}})^2 + \text{const}$.
(One can set const $= \nu(\log\|a_{\text{scale}}\| - \frac{1}{2})$ to match the function values at a_{scale}, but this value is irrelevant for the least squares minimization.)
3) Solve the resulting linear least squares problem in $\mathbf{A}$.
4) Remove any components $\mathbf{a}$ that have become zero, update the scale estimates $a_{\text{scale}} = \|\mathbf{a}\|$, and continue from 2 until convergence.

Fig. 4. An outline of our RVM training algorithm.

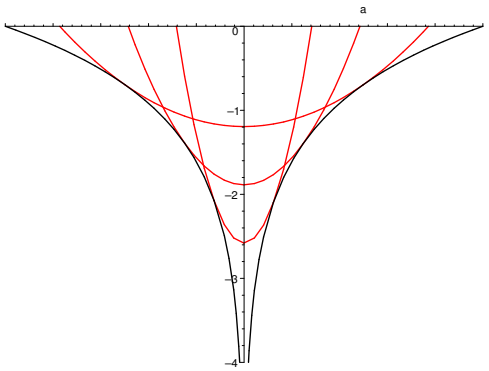


Fig. 5. "Quadratic bridge" approximations to the $\nu \log \|a\|$ regularizers. These are introduced to prevent parameters from prematurely becoming *trapped* at zero. (See text.)

scale hyperparameter. Integrating out the hyperpriors (which can be done analytically) gives singular, highly nonconvex total priors of the form $p(a) \sim \|a\|^{-\nu}$ for each parameter or parameter group a, where ν is a hyperprior parameter. Taking log likelihoods gives an equivalent regularization penalty of the form $R(a) = \nu \log \|a\|$. Note the effect of this penalty. If $\|a\|$ is large, the 'regularizing force' $dR/da \sim \nu/\|a\|$ is small so the prior has little effect on a. But the smaller $\|a\|$ becomes, the greater the regularizing force becomes. At a certain point, the data term no longer suffices to hold the parameter at a nonzero value against this force, and the parameter rapidly converges to zero. Hence, the fitted model is sparse — the RVM automatically selects a subset of 'relevant' basis functions that suffices to describe the problem. The regularizing effect is invariant to rescalings of $\mathbf{f}()$ or $\mathbf{Y}$. (*E.g.* scaling $\mathbf{f} \to \alpha\mathbf{f}$ forces a rescaling $\mathbf{A} \to \mathbf{A}/\alpha$ with no change in residual error, so the regularization forces $1/\|a\| \propto \alpha$ track the data-term gradient $\mathbf{A}\,\mathbf{F}\,\mathbf{F}^\top \propto \alpha$ correctly). ν serves both as a sparsity parameter and as a scale-free regularization parameter. The complete RVM model is highly nonconvex with many local minima and optimizing it can be problematic because relevant parameters can easily become accidentally 'trapped' in the singularity at zero. However, in practice this does not prevent RVMs from giving useful results. Setting ν to optimize the estimation error on a validation set, one typically finds that RVMs give sparse regressors with performance very similar to the much denser ones from analogous methods with milder priors.

To train our RVMs, we do not use Tipping's algorithm [29], but rather a continuation method based on successively approximating the $\nu \log \|a\|$ regularizers with quadratic "bridges" $\nu\,(\|a\|/a_{\text{scale}})^2$ chosen to match the prior gradient at a_{scale}, a running scale estimate for a (see fig. 5). The bridging changes the apparent curvature if the cost surfaces, allowing parameters to pass through zero if they need to, with less risk of premature trapping. The algorithm is sketched in figure 4.

We have tested both *componentwise* priors, $R(\mathbf{A}) = \nu \sum_{jk} \log |\mathbf{A}_{jk}|$, which effectively allow a different set of relevant basis functions to be selected for each dimension of $\mathbf{y}$, and *columnwise* ones, $R(\mathbf{A}) = \nu \sum_k \log \|\mathbf{a}_k\|$ where $\mathbf{a}_k$ is the k^{th} column of $\mathbf{A}$, which select a common set of relevant basis functions for all components of $\mathbf{y}$. Both priors give similar results, but one of the main advantages of sparsity is in reducing the number of basis functions (support features or examples) that need to be evaluated, so in the experiments shown we use columnwise priors. Hence, we minimize

$$\|\mathbf{A}\,\mathbf{F} - \mathbf{Y}\|^2 + \nu \sum_k \log \|\mathbf{a}_k\| \tag{5}$$

C. Choice of Basis

We tested two kinds of regression bases $\mathbf{f}(\mathbf{x})$. *(i) Linear bases*, $\mathbf{f}(\mathbf{x}) \equiv \mathbf{x}$, simply return the input vector, so the regressor is linear in $\mathbf{x}$ and the RVM selects relevant *features* (components of $\mathbf{x}$). *(ii) Kernel bases*, $\mathbf{f}(\mathbf{x}) = (K(\mathbf{x}, \mathbf{x}_1) \;\cdots\; K(\mathbf{x}, \mathbf{x}_n))^\top$, are based on a kernel function $K(\mathbf{x}, \mathbf{x}_i)$ instantiated at training examples $\mathbf{x}_i$, so the RVM effectively selects relevant *examples*. Our experiments with various kernels and combinations of kernels and linear functions show that kernelization (of our already highly non linear features) gives a small but useful improvement in performance — about $0.8°$ per body angle, out of a total mean error of around $7°$. The form and parameters of the kernel have remarkably little influence. The experiments shown use a Gaussian kernel $K(\mathbf{x}, \mathbf{x}_i) = e^{-\beta\|\mathbf{x}-\mathbf{x}_i\|^2}$ with β estimated from the scatter matrix of the training data, but other β values within a factor of 2 from this value give very similar results.

IV. Pose from Static Images

We conducted experiments using a database of motion capture data for a 54 d.o.f. body model (3 angles for each of 18 joints, including body orientation w.r.t the camera). We report mean (over all 54 angles) RMS absolute difference errors between the true and estimated joint angle vectors, in degrees:

$$D(\mathbf{x}, \mathbf{x}') = \frac{1}{m} \sum_{i=1}^{m} |(x_i - x'_i) \bmod \pm 180°| \tag{6}$$

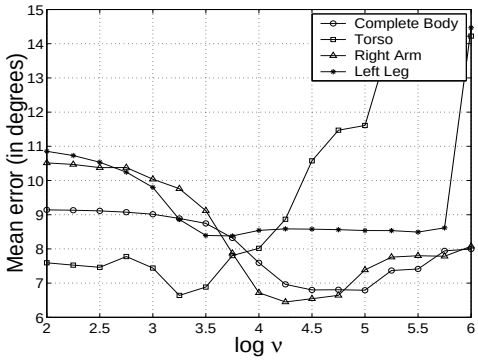


Fig. 6. Mean test-set fitting error for different combinations of body parts, versus the linear RVM spareseness parameter ν. The minima indicate the optimal sparsity / regularization settings for each body part. Limb regressors are sparser than body or torso ones: the whole body regressor retains 23 features; torso, 31; right arm, 10; and the left leg, 7.

Fig. 7. Silhouette points whose shape context classes are retained by the RVM for regression on (a) left arm angles, (b) right leg angles, shown on a sample silhouette. (c-f): Silhouette points encoding torso & neck parameter values over different view points and poses. On average, about 10 features covering about 10% of the silhouette suffice to estimate the pose of each body part.

The training silhouettes were created by using POSER to render the motion captured poses, and reduced to 100-D histograms by vector quantizing their shape context distributions using centres selected by k-means.

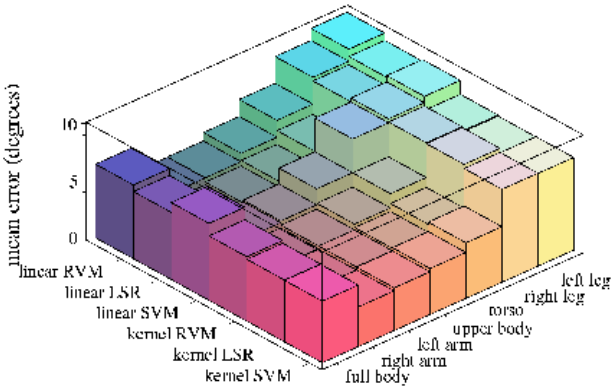
We compare here results of regressing body pose $\mathbf{x}$ (after transforming from 54-D to 55-D as described in section II-B) on the silhouette descriptors $\mathbf{z}$ using ridge, RVM and SVM [32] based regression methods on linear and kernel bases with the functional form given in section III:

$$\mathbf{x} \;=\; \mathbf{A}\,\mathbf{f}(\mathbf{z}) + \boldsymbol{\epsilon} \;\equiv\; \sum_{k=1}^{p} \mathbf{a}_k\,\phi_k(\mathbf{z}) + \boldsymbol{\epsilon} \qquad (7)$$

Ridge regression and RVM regression use quadratic loss functions to measure $\mathbf{x}$-space prediction errors, as described in section III, while SVM regression uses the ϵ-insensitive loss function [26] and a linear programming method for training. The results shown here use the SVM-*light* [15] for implementation.

A. Implicit Feature Selection

Kernel based RVM regression gives reliable pose estimates while retaining only about 6% of the training examples, but working in kernel space hides information associated with individual input features (components of $\mathbf{z}$-vectors). Conversely, linear-basis RVM regression ($\mathbf{f}(\mathbf{z}) = \mathbf{z}$) provides less flexible modelling of the relationship between $\mathbf{x}$ and $\mathbf{z}$, but reveals which of the original input features encode useful pose information, as the RVM directly selects relevant components of $\mathbf{z}$.



	LSR	RVM	SVM
Average error (in degrees)	5.95	6.02	5.91
% of support vectors retained	100	6	53

Fig. 8. (*Top*) A summary of our various regressors' performance on different combinations of body parts for the spiral walking test sequence. (*Bottom*) Error measures for the full body using Gaussian kernel bases with the corresponding number of support vectors retained.

One might expect that, *e.g.* the pose of the arms was mainly encoded by (shape-context classes receiving contributions from) features on the arms, and so forth, so that the arms could be regressed from fewer features than the whole body, and could be regressed robustly even if the legs were occluded. To test this, we divided the body joints into five subsets — torso & neck, the two arms, and the two legs — and trained separate linear RVM regressors for each subset. Fig. 6 shows that similar validation-set errors are attained for each part, but the optimal regularization level is significantly smaller (there is less sparsity) for the torso than for the other parts. Fig. 7 shows the silhouette points whose contexts contribute to the features (histogram classes) that were selected as relevant, for several parts and poses. The two main observations are that the regressors are indeed sparse — only about 10 of the 100 histogram bins were classed as relevant on average, and the points contributing to these tend to be well localized in important-looking regions of the silhouette — but that there is a good deal of non-locality between the points selected for making observations and the parts of the body being estimated. This nonlocality is somewhat surprising. It is perhaps only due to the extent to which the motions of different body segments are synchronized during natural walking motion, but if it turns out to be true for larger training sets containing less orchestrated motions, it may suggest that the localized calculations of model-based pose recovery actually miss a good deal of the information most relevant for pose.

B. Performance Analysis

Fig. 8 summarizes the test-set performance of the various regression methods studied — kernelized and linear basis versions of damped least squares regression (LSR), RVM and SVM regression, for the full body model and various subsets

Fig. 9. Some sample pose reconstructions for a spiral walking sequence not included in the training data. The reconstructions were computed with a Gaussian kernel RVM, using only 156 of the 2636 training examples. The mean angular error per d.o.f. over the whole sequence is $6.0°$. While (a-c) show accurate reconstructions, (d-f) are examples of misestimation: (d) illustrates a label confusion (the left and right legs have been interchanged), (e,f) are examples of compromised solutions where the regressor has averaged between two or more distinct possibilities. Using single images alone, we find $\sim 15\%$ of our results are misestimated.

of it — at optimal regularizer settings computed using 2-fold cross validation. All output parameters are normalized to have unit variance before regression and the tube width ϵ in the SVM is set to correspond to an error of $1°$ for each joint angle. Kernelization brings only a small advantage ($0.8°$ on an average) over purely linear regression against our (highly non-linear) descriptor set. The regressors are all found to give their best results at similar optimal kernel parameters, which are more or less independent of the regularization prior strengths. The RVM regression gives very slightly higher errors than the other two regressors, but much more sparsity. For example, in our whole-body method, the final RVM selects just 156 (about 6%) of the 2636 training points as basis kernels, to give a mean test-set error of $6.0°$. We attribute the slightly better performance of the SVM to the different form of its loss function. The overall similarity of the results obtained from the 3 different regressors confirms that our representation and framework are insensitive to the exact method of regression used.

Fig. 9 shows some sample pose estimation results, on silhouettes from a spiral-walking motion capture sequence that was not included in the training set. The mean estimation error over all joints for the Gaussian RVM in this test is $6.0°$, but the error for individual joints varies depending on the range and discernibility of each joint angle. The RMS errors obtained for some key body angles are as follows (the ranges of variation of these angles in the test set are given in parentheses): body heading angle: $17°$ ($360°$), left shoulder angle: $7.5°$ ($50.8°$), and right hip angle: $4.2°$ ($47.4°$). Fig. 10 (top) plots the estimated and actual values of the overall body heading angle θ during the test sequence, showing that much of the error is due to occasional large errors that we will refer to as "glitches". These are associated with ambiguous cases where the silhouette might easily arise from any of several possible poses. As one diagnostic for this, recall that to allow for the $360°$ wrap around of the heading angle θ, we actually regress $(a, b) = (\cos\theta, \sin\theta)$ rather than θ. In ambiguous cases, the regressor tends to compromise between several possible solutions, and hence returns an (a, b) vector whose norm is significantly less than one. These events are strongly correlated with large estimation errors in θ, as illustrated in fig. 10.

Fig. 11 shows reconstruction results on some real images. The reconstruction quality demonstrates the method's robustness to imperfect visual features, as a quite naive background subtraction method was used to extract somewhat imperfect body silhouettes from these images. The last example demonstrates the problem of silhouette ambiguity: the method returns a pose with the left knee bent instead of the right one as the silhouette looks the same in the two cases, causing a glitch in the output pose.

Although numerically our results are already significantly better than others presented in the literature ($6°$ as compared to RMS errors of about $20°$ per d.o.f. reported in [22]), our pose reconstructions do still contain a significant amount of temporal jitter, and also occasional glitches. The jitter is to be expected given that each image is processed independently. It can be reduced by temporal filtering (simple smoothing or Kalman filtering), and also by adding a temporal dimension to the regressor. The glitches occur when more than one solution is possible, causing the regressor to either 'select' the wrong solution, or to output a compromised solution, different from each. One possible way to reduce such errors would be to incorporate stronger features such as internal body edges within the silhouette, however the problem is bound to persist as important internal body edges are often not visible and useful body edges have to be distinguished from irrelevant clothing texture edges. Furthermore, even without these limb labelling ambiguities, depth related ambiguities continue to remain an issue. By relying on experimentally observed poses, our single image method has already reduced this ambiguity significantly, but human beings often rely on very subtle cues to disambiguate multiple solutions.

In the absence of multiple simultaneous views, temporal continuity is an important supplementary source of information for resolving these ambiguities. In the following two sections, we describe two different approaches that exploit continuity within our regression model.

V. Tracking and Regression

This section describes a novel 'discriminative' tracking framework that fuses pose predictions from a learned dynamical model into our single image regression framework, to correctly reconstruct the most likely 3D pose at each time step. The 3D pose can only be observed indirectly via ambiguous and noisy image measurements, so it is appropriate to start by considering the Bayesian tracking framework in which our knowledge about the state (pose) $\mathbf{x}_t$ given the observations up to time t is represented by a probability distribution, the posterior state density $p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{z}_{t-1}, \ldots, \mathbf{z}_0)$.

Given an image observation $\mathbf{z}_t$ and a prior $p(\mathbf{x}_t)$ on the corresponding pose $\mathbf{x}_t$, the posterior likelihood for $\mathbf{x}_t$ is usu-

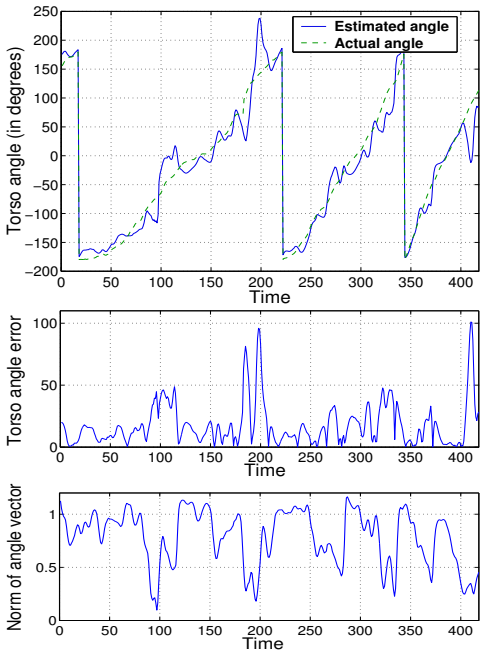


Fig. 10. (Top): The estimated body heading (azimuth θ) over 418 frames of the spiral walking test sequence, compared with its actual value from motion capture. (Middle, Bottom): Episodes of high estimation error are strongly correlated with periods when the norm of the $(\cos\theta, \sin\theta)$ vector that was regressed to estimate θ becomes small. These occur when similar silhouettes arise from very different poses, so that the regressor is forced into outputing a compromise solution.

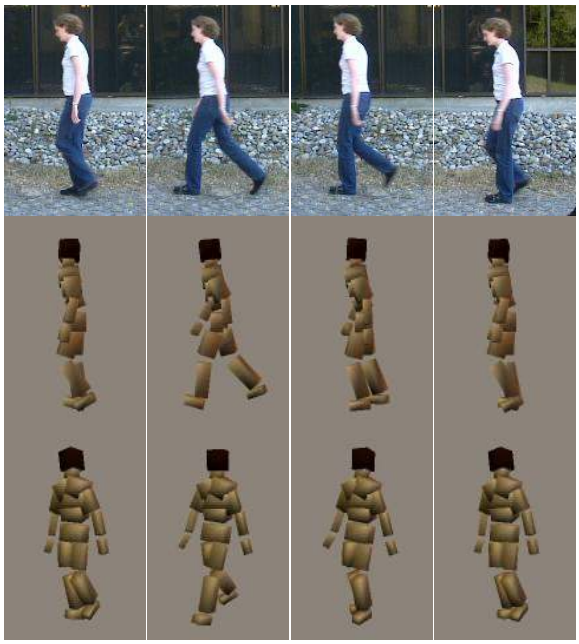


Fig. 11. 3D poses reconstructed from some real test images using a single image for each reconstruction (the images are part of a sequence from www.nada.kth.se/∼hedvig/data.html). The middle and lower rows respectively show the estimates from the original viewpoint and from a new one. The first two columns show accurate reconstructions. In the third column, a noisy silhouette causes slight misestimation of the lower right leg, while the final column demonstrates a case of left-right ambiguity in the silhouette.

ally evaluated using Bayes' rule, $p(\mathbf{x}_t|\mathbf{z}_t) \propto p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t)$, where $p(\mathbf{z}_t|\mathbf{x}_t)$ is an explicit 'generative' observation model that predicts $\mathbf{z}_t$ and its uncertainty given $\mathbf{x}_t$. Unfortunately, when tracking objects as complicated as the human body, the observations depend on a great many factors that are difficult to control, ranging from lighting and background to body shape and clothing style and texture, so any hand-built observation model is necessarily a gross oversimplification. One way around this would be to learn the generative model $p(\mathbf{z}|\mathbf{x})$ from examples, then to work backwards via its Jacobian to get a linearized state update, as in the extended Kalman filter. However, this approach is somewhat indirect, and it may waste a considerable amount of effort modelling appearance details that are irrelevant for predicting pose. Instead, we prefer to learn a 'diagnostic' (discriminative or regressive) model $p(\mathbf{x}|\mathbf{z})$ for the pose $\mathbf{x}$ given the observations $\mathbf{z}$ — *c.f.* the difference between generative and discriminative classifiers, and the regression based trackers of [16, 33]. Similarly, in the context of maximum likelihood pose estimation, we prefer to learn a diagnostic regressor $\mathbf{x} = \mathbf{x}(\mathbf{z})$, *i.e.* a point estimator for the most likely state $\mathbf{x}$ given the observations $\mathbf{z}$, not a generative predictor $\mathbf{z} = \mathbf{z}(\mathbf{x})$. Unfortunately, this brings up a second problem. As we have seen in the previous section, image projection suppresses most of the depth (camera-object distance) information and using silhouettes as image observations induces further ambiguities owing to the lack of limb labelling. So the state-to-observation mapping is always many-to-one. These ambiguities make learning to regress $\mathbf{x}$ from $\mathbf{z}$ difficult because the true mapping is actually multi-valued. A single-valued least squares regressor tends to either zig-zag erratically between different training poses, or (if highly damped) to reproduce their arithmetic mean [7], neither of which is desirable.

To reduce the ambiguity, we work incrementally from the previous few states[4] $\mathbf{x}_{t-1}, \ldots$ (*e.g.* as was done in [10]). We adopt the working hypothesis that given a dynamics based estimate $\mathbf{x}_t(\mathbf{x}_{t-1}, \ldots)$ — or any other rough initial estimate $\check{\mathbf{x}}_t$ for $\mathbf{x}_t$ — it will usually be the case that only one of the observation-based estimates is at all likely a posteriori. Thus, we can use the $\check{\mathbf{x}}_t$ value to "select the correct solution" for the observation-based reconstruction $\mathbf{x}_t(\mathbf{z}_t)$. Formally this gives a regressor $\mathbf{x}_t = \mathbf{x}_t(\mathbf{z}_t, \check{\mathbf{x}}_t)$, where $\check{\mathbf{x}}_t$ serves mainly as a key to select which branch of the pose-from-observation space to use, not as a useful prediction of $\mathbf{x}_t$ in its own right. To work like this, the regressor must be well-localized in $\check{\mathbf{x}}_t$, and hence nonlinear. Taking this one step further, if $\check{\mathbf{x}}_t$ is actually a useful estimate of $\mathbf{x}_t$ (*e.g.* from a dynamical model), we can use a single regressor of the same form, $\mathbf{x}_t = \mathbf{x}_t(\mathbf{z}_t, \check{\mathbf{x}}_t)$, but now with a stronger dependence on $\check{\mathbf{x}}_t$, to capture the net effect of implicitly reconstructing an observation-estimate $\mathbf{x}_t(\mathbf{z}_t)$ and then fusing it with $\check{\mathbf{x}}_t$ to get a better estimate of $\mathbf{x}_t$.

A. Learning the Regression Models

Our discriminative tracking framework now has two levels of regression. We formulate the models as follows and continue to use the methods described in section III:

[4] As an alternative we tried regressing the pose $\mathbf{x}_t$ against a sequence of the last few silhouettes $(\mathbf{z}_t, \mathbf{z}_{t-1}, \ldots)$, but the ambiguities are found to persist for several frames.

1) Dynamical (Prediction) Model: Human body dynamics can be modelled fairly accurately with a second order linear autoregressive process, $\mathbf{x}_t = \check{\mathbf{x}}_t + \boldsymbol{\epsilon}$, where $\check{\mathbf{x}}_t \equiv \tilde{\mathbf{A}}\,\mathbf{x}_{t-1} + \tilde{\mathbf{B}}\,\mathbf{x}_{t-2}$ is the second order dynamical estimate of $\mathbf{x}_t$ and $\boldsymbol{\epsilon}$ is a residual error vector (*c.f. e.g.* [3]). To ensure dynamical stability and avoid over-fitting, we actually learn the autoregression for $\check{\mathbf{x}}_t$ in the following form:

$$\check{\mathbf{x}}_t \;\equiv\; (\mathbf{I}+\mathbf{A})(2\mathbf{x}_{t-1}-\mathbf{x}_{t-2}) + \mathbf{B}\,\mathbf{x}_{t-1} \tag{8}$$

where $\mathbf{I}$ is the $m \times m$ identity matrix. This form helps to maintain stability by converging towards a default linear prediction if $\mathbf{A}$ and $\mathbf{B}$ are overdamped. We estimate $\mathbf{A}$ and $\mathbf{B}$ by regularized least squares regression against $\mathbf{x}_t$, minimizing $\|\boldsymbol{\epsilon}\|_2^2 + \lambda(\|\mathbf{A}\|_{\text{Frob}}^2 + \|\mathbf{B}\|_{\text{Frob}}^2)$ over the training set, with the regularization parameter λ set by cross-validation to give a well-damped solution with good generalization.

2) Likelihood (Correction) Model: Now consider the observation model. As discussed above, the underlying density $p(\mathbf{x}_t \,|\, \mathbf{z}_t)$ is highly multimodal owing to the pervasive ambiguities in reconstructing 3D pose from monocular images, so no single-valued regression function $\mathbf{x}_t = \mathbf{x}_t(\mathbf{z}_t)$ can give acceptable point estimates for $\mathbf{x}_t$. However much of the 'glitchiness' and jitter observed in the static reconstructions of section IV-B can be removed by feeding $\check{\mathbf{x}}_t$ into the regression model. The combined regressor can be formulated in several different ways. The simplest is to linearly combine $\check{\mathbf{x}}_t$ with the estimate $\mathbf{x}_t$ given by equation (7), but this only smooths the results, reducing jitter, while still continuing to give wrong solutions when (7) returns a wrong estimate. We thus include a non-linear dependence on $\check{\mathbf{x}}_t$ with $\mathbf{z}_t$ in the observation-based regressor, giving a state sensitive observation update. Our full regression model also includes an explicit linear $\check{\mathbf{x}}_t$ term to represent the direct contribution of the dynamics to the overall state estimate, so the final model becomes $\mathbf{x}_t \equiv \hat{\mathbf{x}}_t + \boldsymbol{\epsilon}'$ where $\boldsymbol{\epsilon}'$ is a residual error to be minimized, and:

$$\hat{\mathbf{x}}_t \;=\; \mathbf{C}\,\check{\mathbf{x}}_t + \sum_{k=1}^{p} \mathbf{d}_k\,\phi_k(\check{\mathbf{x}}_t, \mathbf{z}_t) \;\equiv\; \begin{pmatrix}\mathbf{C} & \mathbf{D}\end{pmatrix} \begin{pmatrix}\check{\mathbf{x}}_t \\ \mathbf{f}(\check{\mathbf{x}}_t, \mathbf{z}_t)\end{pmatrix} \tag{9}$$

Here, $\{\phi_k(\mathbf{x},\mathbf{z}) \,|\, k = 1 \ldots p\}$ is a set of scalar-valued non-linear basis functions for the regression, and $\mathbf{d}_k$ are the corresponding $\mathbb{R}^m$-valued weight vectors. For compactness, we gather these into an $\mathbb{R}^p$-valued feature vector $\mathbf{f}(\mathbf{x},\mathbf{z}) \equiv (\phi_1(\mathbf{x},\mathbf{z}), \ldots, \phi_p(\mathbf{x},\mathbf{z}))^\top$ and an $m \times p$ weight matrix $\mathbf{D} \equiv (\mathbf{d}_1, \ldots, \mathbf{d}_p)$. In the experiments reported here, we used instantiated-kernel bases of the form

$$\phi_k(\mathbf{x},\mathbf{z}) = K_x(\mathbf{x},\mathbf{x}_k) \cdot K_z(\mathbf{z},\mathbf{z}_k) \tag{10}$$

where $(\mathbf{x}_k, \mathbf{z}_k)$ is a training example and K_x, K_z are (here, independent Gaussian) kernels on $\mathbf{x}$-space and $\mathbf{z}$-space, $K_x(\mathbf{x},\mathbf{x}_k) = e^{-\beta_x \|\mathbf{x}-\mathbf{x}_k\|^2}$ and $K_z(\mathbf{z},\mathbf{z}_k) = e^{-\beta_z \|\mathbf{z}-\mathbf{z}_k\|^2}$. Building the basis from Gaussians based at training examples in joint $(\mathbf{x},\mathbf{z})$ space makes examples relevant only if they have similar image silhouettes *and* similar underlying poses.

Mistracking due to extinction. Kernelization in joint $(\mathbf{x},\mathbf{z})$ space allows the relevant branch of the inverse solution to be chosen, but it is essential to choose the relative widths of the kernels appropriately. If the $\mathbf{x}$-kernel is chosen too wide, the method tends to average over (or zig-zag between) several alternative pose-from-observation solutions, which defeats the purpose of including $\check{\mathbf{x}}$ in the observation regression. On the other hand, too much locality in $\mathbf{x}$ effectively 'switches off' the observation-based state corrections whenever the estimated state happens to wander too far from the observed training examples $\mathbf{x}_k$. So if the $\mathbf{x}$-kernel is set too narrow, observation information is only incorporated sporadically and mistracking can easily occur. Fig. 12 illustrates this effect, for an $\mathbf{x}$-kernel a factor of 10 narrower than the optimum. The method initially seemed to be sensitive to the kernel width parameters, but after fixing good default values by cross-validation on an independent motion sequence we observed accurate performance over a sufficiently wide range for both the kernel widths: a tolerance factor of about 2 on β_x and about 4 on β_z.

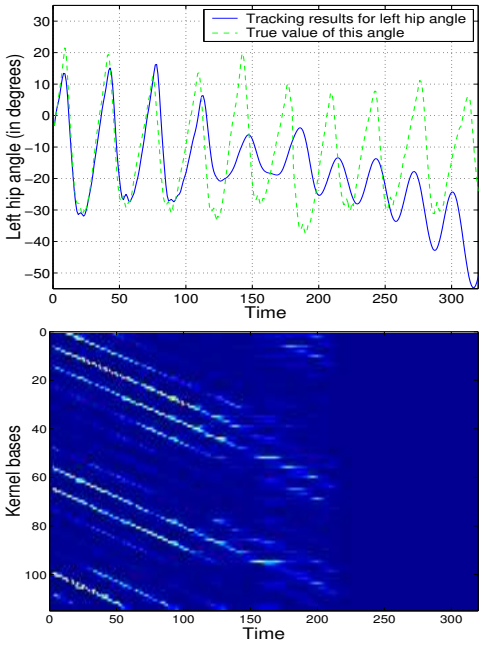


Fig. 12. An example of mistracking caused by an over-narrow pose kernel K_x. The kernel width is set to 1/10 of the optimal value, causing the tracker to lose track from about t=120, after which the state estimate drifts away from the training region and all kernels stop firing by about t=200. *Left:* the variation of a left hip angle parameter for a test sequence of a person walking in a spiral. *Right:* The temporal activity of the 120 kernels (training examples) during this track. The banded pattern occurs because the kernels are samples taken from along a similar 2.5 cycle spiral walking sequence, each circuit involving about 8 steps. The similarity between adjacent steps and between different circuits is clearly visible, showing that the regressor can locally still generalize well.

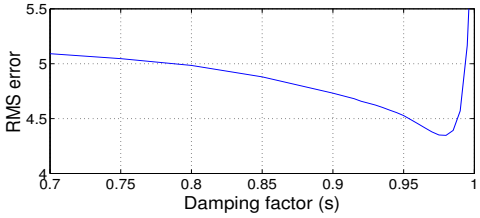


Fig. 13. The variation of the RMS test-set tracking error with damping factor s. See the text for discussion.

Neutral vs Damped Dynamics. The coefficient matrix $\mathbf{C}$ in (9) plays an interesting role. Setting $\mathbf{C} \equiv \mathbf{I}$ forces the correction model to act as a differential update on $\check{\mathbf{x}}_t$ (what we refer to as having a 'neutral' dynamical model). On the other extreme, $\mathbf{C} \equiv \mathbf{0}$ gives largely observation-based state estimates with only a latent dependence on the dynamics. An intermediate setting, however, turns out to give the best overall results. Damping the dynamics slightly ensures stability and controls drift — in particular, preventing the observations from disastrously 'switching off' because the state has drifted too far from the training examples — while still allowing a reasonable amount of dynamical smoothing. Usually we estimate the full (regularized) matrix $\mathbf{C}$ from the training data, but to get an idea of the trade-offs involved, we also studied the effect of explicitly setting $\mathbf{C} = s\mathbf{I}$ for $s \in [0, 1]$. We find that a small amount of damping, $s_{opt} \approx .98$ gives the best results overall, maintaining a good lock on the observations without losing too much dynamical smoothing (see fig. 13.) This simple heuristic setting gives very similar results to the model obtained by learning the full matrix $\mathbf{C}$.

B. Tracking Results

We trained the new regression model (9) on our motion capture data as in section IV. For these experiments, we used 8 different sequences totalling about 2000 instantaneous poses for training, and another two sequences of about 400 points each as validation and test sets. Errors are again reported as described by (6).

The dynamical model is learned from the training data exactly as described in §V-A.1, but when training the observation model, we find that its coverage and capture radius can be increased by including a wider selection of $\check{\mathbf{x}}_t$ values than those produced by the dynamical predictions. Hence, we train the model $\mathbf{x} = \mathbf{x}_t(\check{\mathbf{x}}, \mathbf{z})$ using a combination of 'observed' samples $(\check{\mathbf{x}}_t, \mathbf{z}_t)$ (with $\check{\mathbf{x}}_t$ computed from (8)) and artificial samples generated by Gaussian sampling $\mathcal{N}(\mathbf{x}_t, \Sigma)$ around the training state $\mathbf{x}_t$. The observation $\mathbf{z}_t$ corresponding to $\mathbf{x}_t$ is still used, forcing the observation based part of the regressor to rely mainly on the observations, *i.e.* on recovering $\mathbf{x}_t$ (or at least an update to $\check{\mathbf{x}}_t$) from $\mathbf{z}_t$, using $\check{\mathbf{x}}_t$ mainly as a hint about the inverse solution to choose. The covariance matrix Σ is chosen to reflect the local scatter of the training examples, with a larger variance along the tangent to the trajectory at each point to ensure that phase lag between the state estimate and the true state is reliably detected and corrected.

Fig. 14 illustrates the relative contributions of the dynamics and observation terms in our model by plotting tracking results for a motion capture test sequence in which the subject walks in a decreasing spiral. This sequence was not included in the training set, although similar ones were. The purely dynamical model (8) provides good estimates for a few time steps, but gradually damps and drifts out of phase. Such damped oscillations are characteristic of second order linear autoregressive dynamics, trained with enough regularization to ensure model stability. The results based on observations alone without any temporal information are included again here for comparison. These are obtained from (7), which is actually a special case of (9) where $\mathbf{C} = \mathbf{0}$ and $K_x = 1$. Panels (c),(f) show that jointly regressing dynamics and observations gives a significant improvement in estimation quality, with smoother and stabler tracking. There is still some residual misestimation of the hip angle in (c) at around t=140 and t=380. At these points, the subject is walking directly towards the camera (heading angle $\theta \sim 0°$), so the only cue for hip angle is the position of the corresponding foot, which is sometimes occluded by the opposite leg. Humans also find it difficult to estimate this angle from the silhouette at these points.

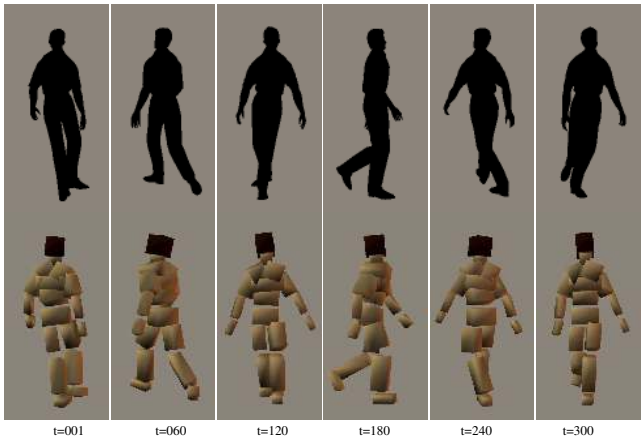


Fig. 15. Sample pose reconstructions for the spiral walking sequence using the tracking method. This sequence was not included in the training data, and corresponds to figures 14(c) & (f). The reconstructions were computed with a Gaussian kernel RVM, using only 18% training examples. The average RMS estimation error per d.o.f. over the whole sequence is $4.1°$.

Fig. 15 shows some silhouettes and corresponding maximum likelihood pose reconstructions, for the same test sequence. The 3D poses for the first two time steps were set by hand to initialize the dynamical predictions. The average RMS estimation error over all joints using the RVM regressor in this test is $4.1°$. Well-regularized least squares regression over the same basis gives similar errors, but has much higher storage requirements. The Gaussian RVM gives a sparse regressor for (9) involving only 348 of the 1927 (18%) training examples, thus allowing a significant reduction in the amount of training data that needs to be stored. The reconstruction results on two test video sequences are shown in figs 16 and 19.

In terms of computation time, the final RVM regressor already runs in real time in Matlab. Silhouette extraction and shape-context descriptor computations are currently done offline, but should be feasible online in real time. The offline learning process takes about 2-3 min for the RVM with $\sim$2000 data points, and currently about 20 min for Shape Context extraction and clustering (this being highly unoptimized Matlab code).

Automatic Initialization: The method is reasonably robust to initialization errors. Although the results shown in figs. 14 and 15 were obtained by initializing from ground truth, we also tested the effects of automatic (and hence potentially incorrect) initialization. In an experiment in which the tracker was automatically initialized at each time step in turn using the pure observation model, then tracked forwards and back-

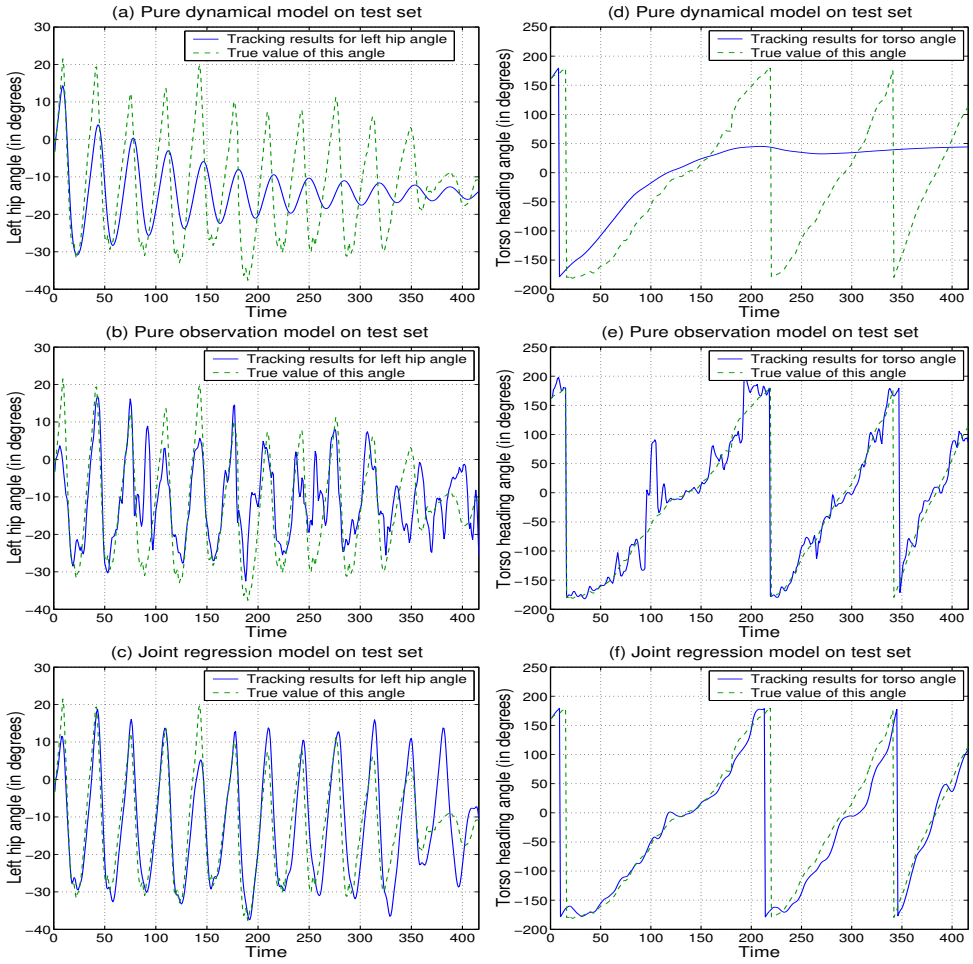


Fig. 14. Sample tracking results on a spiral walking test sequence. (a) Variation of the left hip-angle parameter, as predicted by a pure dynamical model initialized at $t = \{0, 1\}$, (b) Estimated values of this angle from regression on observations alone (*i.e.* no initialization or temporal information), (c) Results from our novel joint regressor, obtained by combining dynamical and state+observation based regression models. (d,e,f) Similar plots for the overall body rotation angle. Note that this angle wraps around at 360°, i.e. $\theta \simeq \theta \pm 360^\circ$.

wards using the dynamical tracker, the initialization lead to successful tracking in 84% of the cases. The failures were the 'glitches', where the observation model gave completely incorrect initializations.

VI. Resolving Ambiguities using a Mixture of Experts

In this section, we discuss an alternative approach to dealing with multiple possible solutions in the 3D pose estimation problem. We extend our single image regression framework from section IV to a *mixture* of regressors (often known as a mixture of experts [14]). Such a model enables the regressor to output more than one possible solution from a single silhouette — in general a multimodal probability density $p(\mathbf{x}|\mathbf{z})$. We describe the formulation of our mixture model and show how it can be used in a multiple hypothesis probabilistic tracking framework to achieve smooth reconstruction tracks free from glitches.

A. Probabilistic pose from static images

A close analysis of the nature of ambiguities in the silhouette-to-pose problem indicates that they are of more than one type in nature. Firstly, there exist instances where any 3D pose in a continuous range seems to explain the given silhouette observation quite well, *e.g.* estimating out-of-plane rotations where the limb length signal is not strong enough to estimate the angle accurately. Here one would desire a broad distribution in 3D pose space as the output from a single silhouette. Other cases of ambiguity arise due to kinematic flipping (*c.f.* [24]) or label-ambiguities (disambiguating the left and right arms/legs). In such cases, there is typically a finite discrete set of probable solutions — often only 2 or 4, but sometimes more. To deal with both of the above cases, we model the conditional density $p(\mathbf{x}|\mathbf{z})$ as a mixture of Gaussians:

$$p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\bar{\mathbf{x}}_k, \mathbf{\Lambda}_k) \tag{11}$$

where $\bar{\mathbf{x}}_k$ is computed by learning a regressor $\bar{\mathbf{x}}_k = \mathbf{A}_k\, \mathbf{f}(\mathbf{z}) + \mathbf{b}_k$ within each mixture component, and $\mathbf{\Lambda_k}$ (a diagonal covariance matrix in our case) is estimated from residual errors. π_k are the gating probabilities of the regressors. Setting

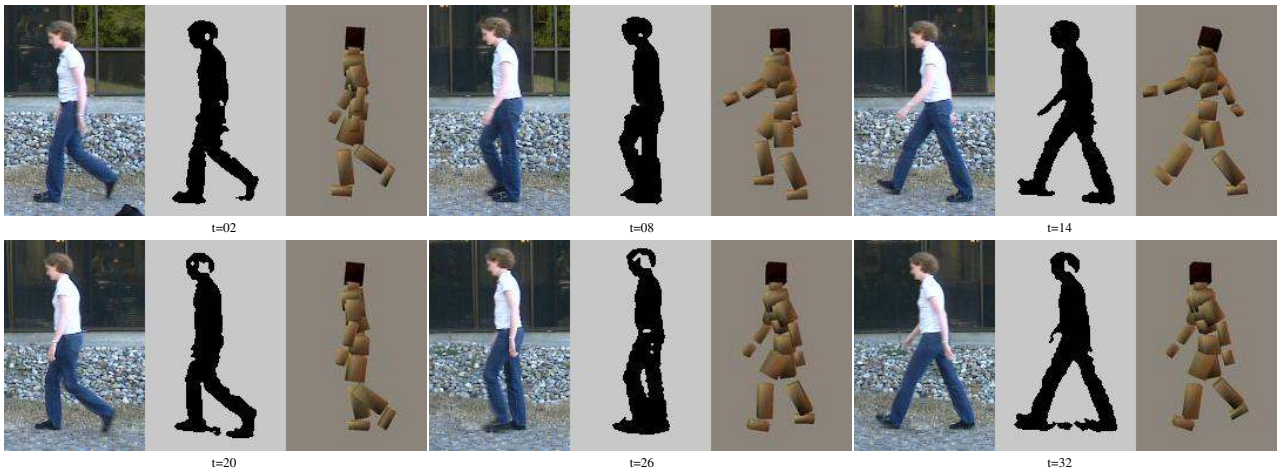


Fig. 16. 3D poses reconstructed from a test video sequence (obtained from www.nada.kth.se/∼hedvig/data.html). The presence of shadows and holes in the extracted silhouettes demonstrates the robustness of our shape descriptors — however, a weak or noisy observation signal sometimes causes failure to track accurately. *E.g.* at $t = 8, 14$, the pose estimates are dominated by the dynamical predictions, which do ensure smooth and natural motion but may cause slight mistracking of some parameters.

$\mathbf{f}(\mathbf{z}) \equiv \mathbf{z}$ simplifies the problem to learning a mixture of *linear* regressors. The model is learned by fitting a mixture of Gaussians to the joint probability density $(\mathbf{z}^\top, \mathbf{x}^\top)^\top$:

$$\begin{pmatrix}\mathbf{z}\\ \mathbf{x}\end{pmatrix} = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Gamma}_k)$$

$$\boldsymbol{\mu}_k = \begin{pmatrix}\bar{\mathbf{z}}_k\\ \mathbf{A}_k\bar{\mathbf{z}}_k + \mathbf{b}_k\end{pmatrix}, \boldsymbol{\Gamma}_k = \begin{pmatrix}\boldsymbol{\Sigma}_k & \boldsymbol{\Sigma}_k\mathbf{A}_k^\top\\ \mathbf{A}_k\boldsymbol{\Sigma}_k & \mathbf{A}_k\boldsymbol{\Sigma}_k\mathbf{A}_k^\top + \boldsymbol{\Lambda}_k\end{pmatrix} \tag{12}$$

To avoid overfitting, we constrain the descriptor covariance matrix $\boldsymbol{\Sigma}$ to be diagonal, thereby drastically reducing the number of parameters to be estimated in our model. The gating probabilities are given by $\pi_k(\mathbf{z}) = \frac{1}{Z}|\boldsymbol{\Sigma}_k|^{-1}e^{-\frac{1}{2}(\mathbf{z}-\bar{\mathbf{z}}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{z}-\bar{\mathbf{z}}_k)}$.

The parameters are learned using a standard Expectation Maximization (EM) algorithm. We initialize the class centers and gating probabilities by clustering in the $\mathbf{x}$-space alone in order to separate points that have similar $\mathbf{z}$-values but different $\mathbf{x}$ values. (Including $\mathbf{z}$ in the initial clustering decreased the quality of separation between ambiguous cases). Results show that most of the ambiguities are resolved and the regressors indeed learn separate models for the multiple possible solutions that come from different regions of the pose space. Figure 17 shows the two most highly weighted modes of the distribution in 3D pose obtained by using a mixture of 8 regressors over some sample silhouettes. These two solutions usually capture the principal ambiguities, but valid reconstructions are often also present in some of the remaining 6 modes of the output.

The associated probabilities of these modes are given by the gating probabilities π_k of the regressors used for the reconstruction. We find that these gating probabilities typically give a good idea of the true number of ambiguous solutions in the given case, but they do not always select the correct solution from among the generated possibilities. To get an idea of the number of cases where the system cannot choose a single 'correct' solution, we rank the various modes obtained by the regressors according to their *(i)* estimated probabilities π_k, and *(ii)* their accuracies obtained by comparison with the ground truth. We find that in 30-35% of the cases, the solution that is estimated as being most *likely* is actually incorrect — but most of these correspond to cases that are truly ambiguous

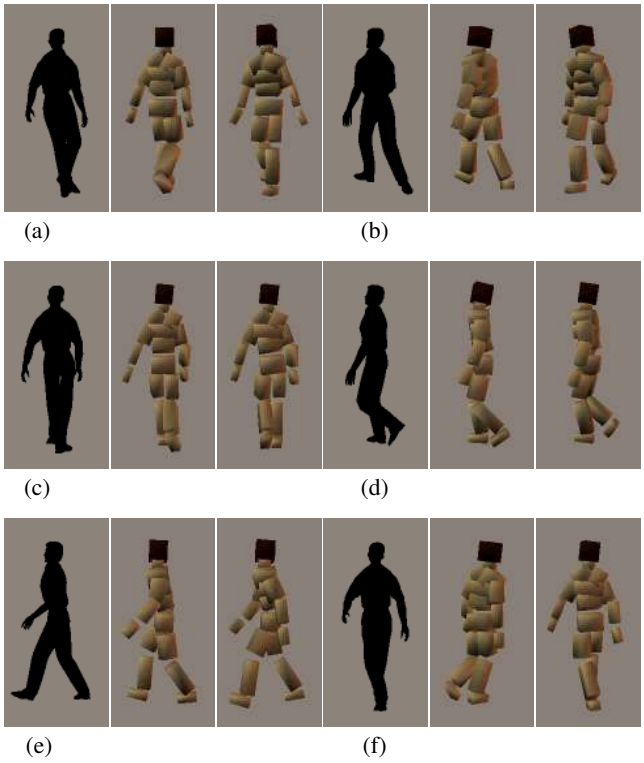


Fig. 17. Multiple possible 3D pose estimates obtained from individual silhouettes using a mixture of regressors. The two most likely modes of the distribution are shown in each case, and generally capture the two most evident reconstruction possibilities, illustrating cases of forward-backward ambiguity (a,b), kinematic flipping of the legs (c) and interchanging labels between the two legs (d,e). (f) shows an example where the first solution is a misestimate but feasible solutions are obtained in the other modes.

— and the correct solution is usually amongst the few most probable ones.

Using a mixture model scheme in place of a single regressor allows most of the instances of compromised solutions from the single regressor to be resolved into several solutions capturing the different 3D possibilities (*e.g.* compare figures 9(e) and 17(e)). This gives the method the capability of accurately estimating possible 3D poses from single images — even in the cases of ambiguity — by outputing several possible solutions whenever they exist. Below we describe how to use these multiple possible solution sets across a sequence of silhouettes to allow smooth tracking free from glitches.

B. Condensation based tracking

The multimodal likelihoods obtained in the previous section can be used in a tracker that combines the modes across time to estimate a temporally coherent maximum likelihood trajectory of 3D poses. This is demonstrated by implementing a CONDENSATION [13] based tracking algorithm that uses the output density of our mixture model to assign likelihoods to its particles. We work with the assumption that state information from the current observation is independent of state information from the dynamics:

$$p(\mathbf{x}_t \,|\, \mathbf{z}_t, \mathbf{x}_{t-1}, \ldots) \;\propto\; p(\mathbf{x}_t \,|\, \mathbf{z}_t)\, p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \ldots) \qquad (13)$$

The pose reconstruction ambiguity is reflected in the fact that the likelihood $p(\mathbf{x}_t|\mathbf{z}_t)$ is typically multimodal. It is often obtained by using Bayes' rule to invert to the many-to-one generative model $p(\mathbf{z}_t|\mathbf{x}_t)$, but we continue to work in our discriminative tracking framework and hence use $p(\mathbf{x}_t|\mathbf{z}_t)$ as opposed to $p(\mathbf{z}_t|\mathbf{x}_t)$. The dynamical model from section V-A.1 is used to generate an estimate of the 3D pose distribution $p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \ldots)$. Samples $(\check{\mathbf{x}}_t^i)$ from this distribution are then assigned weights $p(\check{\mathbf{x}}_t^i \mathbf{z})$ by the observation model density as given in (11).

Figure 18 shows tracking results obtained on our spiral walk test set using CONDENSATION with 2000 particles. In general, the method tracks through the correct modes of the observation density. Smooth tracks are produced, with the maximum likelihood reconstructions usually being more accurate than any of the 8 individual modes of the multimodal regressor output alone.

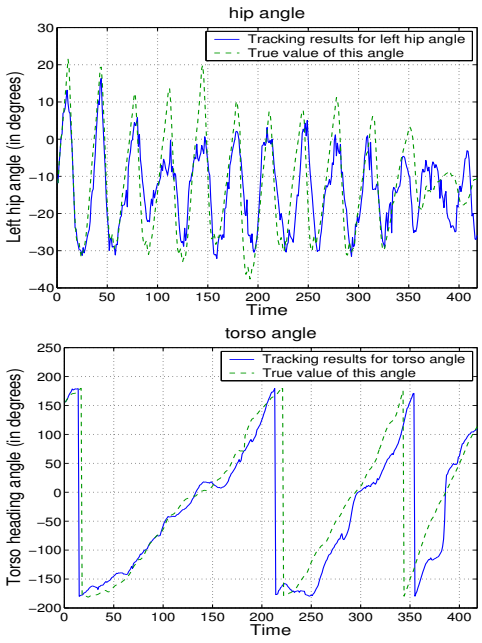


Fig. 18. Tracking results with a particle filter on a spiral walk test sequence using the mixture of regressors output as an observation model: *(Left)* left hip angle parameter, *(Right)* torso heading angle.

VII. DISCUSSIONS AND CONCLUSIONS

We have presented a method that recovers 3D human body pose from monocular silhouettes by direct nonlinear regression of joint angles against histogram-of-shape-context silhouette shape descriptors. Neither a 3D body model nor labelled image positions of body parts are needed, making the method easily adaptable to different people, appearances and representations of 3D human body pose. The regression is done in either linear or kernel space, using either ridge regression or Relevance Vector Machines. The main advantage of RVMs is that they allow sparse sets of highly relevant features or training examples to be selected for the regression. We have proposed two ways of overcoming the intrinsic ambiguity of the pose-from-monocular-observations problem: regressing the pose jointly on image observations and previous pose; and using a mixture of regressors in a multiple hypothesis tracking scheme. Both of these produce stable, temporally consistent tracking. Our mixture of regressors scheme has the capability to reconstruct 3D human pose accurately from a single image, giving multiple possible poses whenever they exist.

Our kernelized RVM regressors retain only about $15-20\%$ of their training examples in the regression based tracking, thus giving a large effective reduction in storage space compared to nearest neighbour methods, which must retain the whole training database. Our methods show promising results, being about three times more accurate than the current state of the art [22].

Future work: We plan to investigate the extension of our regression based system to cover a wider class of human motions and also add structured representations to our model for dealing with greater variability in the 54 dimensional output space. On the vision side, we would like to include richer features, such as internal edges in addition to silhouette boundaries to reduce susceptibility to poor image segmentation.

Our linear RVMs directly select relevant features in the image descriptor space. This property may be useful for identifying better feature sets, not only for pose recovery and tracking, but also for human detection tasks.

ACKNOWLEDGMENTS

This work was supported by the European Union projects VIBES and LAVA, and the research network PASCAL.

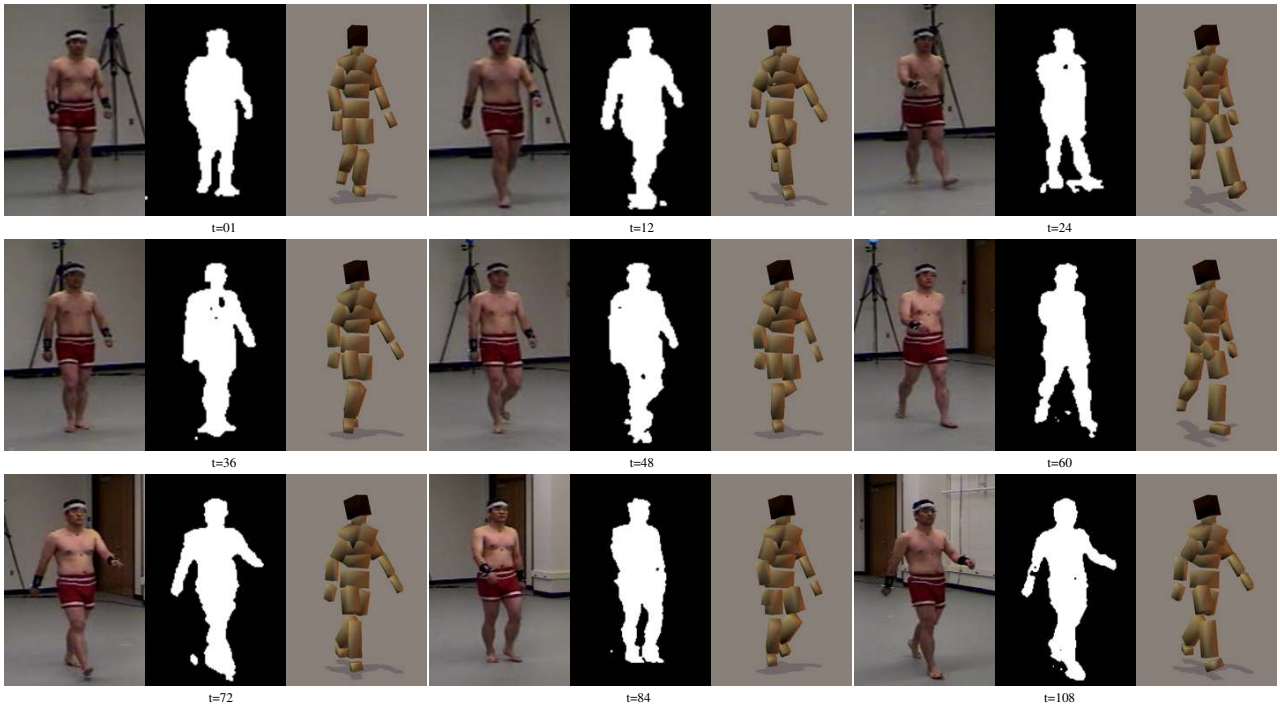


Fig. 19. 3D poses reconstructed from another test video sequence (obtained from http://mocap.cs.cmu.edu/). In this sequence the subject walks towards the camera causing a scale change by a factor of $\sim$2. (The images and silhouettes have been normalized in scale here for display purposes). Our scale invariant silhouette representation allows the algorithm to process a silhouette independent of its size or location in the image without disturbing the 3D pose recovery.

REFERENCES

[1] A. Agarwal and B. Triggs. 3D Human Pose from Silhouettes by Relevance Vector Regression. In *Int. Conf. Computer Vision & Pattern Recognition*, 2004.

[2] A. Agarwal and B. Triggs. Learning to Track 3D Human Motion from Silhouettes. In *Int. Conf. on Machine Learning*, 2004.

[3] A. Agarwal and B. Triggs. Tracking Articulated Motion with Piecewise Learned Dynamical Models. In *European Conf. Computer Vision*, 2004.

[4] V. Athitsos and S. Sclaroff. Inferring Body Pose without Tracking Body Parts. In *Int. Conf. Computer Vision & Pattern Recognition*, 2000.

[5] V. Athitsos and S. Sclaroff. Estimating 3D Hand Pose From a Cluttered Image. In *Int. Conf. Computer Vision*, 2003.

[6] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition using Shape Contexts. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 24(4):509–522, 2002.

[7] C. Bishop. *Neural Networks for Pattern Recognition*, chapter 6. Oxford University Press, 1995.

[8] M. Brand. Shadow Puppetry. In *Int. Conf. Computer Vision*, pages 1237–1244, 1999.

[9] C. Bregler and J. Malik. Tracking People with Twists and Exponential Maps. In *Int. Conf. Computer Vision & Pattern Recognition*, pages 8–15, 1998.

[10] A. D'Souza, S. Vijayakumar, and S. Schaal. Learning Inverse Kinematics. In *Int. Conf. on Intelligent Robots and Systems*, 2001.

[11] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3D Structure with a Statistical Image-Based Shape Model. In *Int. Conf. Computer Vision*, pages 641–648, 2003.

[12] N. Howe, M. Leventon, and W. Freeman. Bayesian Reconstruction of 3D Human Motion from Single-Camera Video. In *Neural Information Processing Systems*, 1999.

[13] M. Isard and A. Blake. CONDENSATION – Conditional Density Propagation for Visual Tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.

[14] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87, 1991.

[15] T. Joachims. Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[16] F. Jurie and M. Dhome. Hyperplane Approximation for Template Matching. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 24(7):996–1000, 2002.

[17] D. Lowe. Object Recognition from Local Scale-invariant Features. In *Int. Conf. Computer Vision*, pages 1150–1157, 1999.

[18] G. Mori and J. Malik. Estimating Human Body Configurations Using Shape Context Matching. In *European Conf. Computer Vision*, volume 3, pages 666–680, 2002.

[19] D. Ormoneit, H. Sidenbladh, M. Black, and T. Hastie. Learning and Tracking Cyclic Human Motion. In *Neural Information Processing Systems*, pages 894–900, 2000.

[20] V. Pavlovic, J. Rehg, and J. MacCormick. Learning Switching Linear Models of Human Motion. In *Neural Information Processing Systems*, pages 981–987, 2000.

[21] Y. Rubner, C. Tomasi, and L.J. Guibas. A Metric for Distributions with Applications to Image Databases. In *Int. Conf. Computer Vision*, Bombay, 1998.

[22] G. Shakhnarovich, P. Viola, and T. Darrell. Fast Pose Estimation with Parameter Sensitive Hashing. In *Int. Conf. Computer Vision*, 2003.

[23] H. Sidenbladh, M. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conf. Computer Vision*, volume 1, 2002.

[24] C. Sminchisescu and B. Triggs. Covariance Scaled Sampling for Monocular 3D Body Tracking. In *Int. Conf. Computer Vision & Pattern Recognition*, 2001.

[25] C. Sminchisescu and B. Triggs. Kinematic Jump Processes For Monocular 3D Human Tracking. In *Int. Conf. Computer Vision & Pattern Recognition*, June 2003.

[26] A. Smola and B. Schölkpof. A Tutorial on Support Vector Regression. *NeuroCOLT2 Technical Report NC2-TR-1998-030*, 1998.
[27] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Filtering Using a Tree-Based Estimator. In *Int. Conf. Computer Vision*, 2003.
[28] C. Taylor. Reconstruction of Articulated Objects from Point Correspondances in a Single Uncalibrated Image. In *Int. Conf. Computer Vision & Pattern Recognition*, 2000.
[29] M. Tipping. The Relevance Vector Machine. In *Neural Information Processing Systems*, 2000.
[30] M. Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. *J. Machine Learning Research*, 1:211–244, 2001.
[31] K. Toyama and A. Blake. Probabilistic Tracking in a Metric Space. In *Int. Conf. Computer Vision*, pages 50–59, 2001.
[32] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
[33] O. Williams, A. Blake, and R. Cipolla. A Sparse Probabilistic Learning Algorithm for Real-Time Tracking. In *Int. Conf. Computer Vision*, 2003.