Recovering 3D human pose from monocular images

Ankur Agarwal, Bill Triggs

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 labeling 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. The loss of depth and limb labeling information often makes the recovery of 3D pose from single silhouettes ambiguous. To handle this, the method is embedded in a novel regressive tracking framework, using dynamics from the previous state estimate together with a learned regression value to disambiguate the pose. We show that the resulting system tracks long sequences stably. 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 for several representations of full body pose, both quantitatively on independent but similar test data and qualitatively on real image sequences. Mean angular errors of 4-6° are obtained for a variety of walking motions.

Index Terms—Computer vision, human motion estimation, machine learning, multivariate regression.

1 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 which requires known image positions for each body part [27]—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 [24]. 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 modeling and rendering, instead capitalizing on the fact that the set of *typical* human poses is far smaller than the set of kinematically possible ones and 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, estimating pose by searching for training image(s) similar to the given input image and interpolating from their poses [5], [18], [22], [26].

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 compact model that has good generalization to unseen examples. Given the high dimensionality and intrinsic ambiguity of the monocular pose estimation problem, active 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 *Relevance Vector Machine (RVM)* [28] regression. RVMs have been used, e.g., to build kernel regressors for 2D displacement updates in correlation-based patch tracking [32]. 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. The loss of depth and limb labeling information often makes the recovery of 3D pose from single silhouettes ambiguous. To overcome this problem, the method is embedded in a tracking framework, combining dynamics from the previous state estimate with a special regressor to disambiguate the pose. Tracking is then formulated either as a single fully regressive model or by using the regression estimates in a multiple hypothesis tracker based on Condensation [13].

1.1 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

• *The authors are with INRIA Rhône-Alpes, 665, Avenue de l'Europe, 38330 Montbonnot, France. E-mail: {Ankur.Agarwal, Bill.Triggs}@inrialpes.fr.*

Manuscript received 11 Sept. 2004; revised 8 Apr. 2005; accepted 26 Apr. 2005; published online 11 Nov. 2005.
Recommended for acceptance by P. Fua.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0479-0904.

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 degrees of freedom 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 degrees of freedom)—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 center of each body joint as an intermediate representation, first estimating these joint centers in the image, then recovering 3D pose from them. Howe et al. [12] develop a Bayesian learning framework to recover 3D pose from known centers, based on a training set of pose-center pairs obtained from resynthesized motion capture data. Mori and Malik [18] estimate the centers using shape context image matching against a set of training images with prelabeled centers, then reconstruct 3D pose using the algorithm of [27]. These approaches show that using 2D joint centers 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.

With regard to 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], [30]. 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 fused directly with pose proposals computed from the observed image. Our algorithm is related to Bayesian tracking, but we use learned regression (inverse) models to eliminate the need for an explicit body model that is projected to predict image observations. A brief description of our single image regression-based scheme is given in [1] and the extension that resolves ambiguities using dynamics first appeared in [2].

1.2 Overview of the Approach

We represent 3D body pose by 55D vectors $\mathbf{x}$ including three joint angles for each of the 18 major body joints. Not all of these degrees of freedom are independent, but they correspond to the motion capture data that we use to train the system (see Section 2.2). The input images are reduced to 100D observation vectors $\mathbf{z}$ that robustly encode the shape of a human image silhouette (Section 2.1). Given a set of labeled training examples $\{(\mathbf{z}_i, \mathbf{x}_i) \mid i = 1 \ldots n\}$, the RVM learns a smooth reconstruction function $\mathbf{x} = \mathbf{r}(\mathbf{z}) = \sum_k \mathbf{a}_k \, \phi_k(\mathbf{z})$ that is valid over the region spanned by the training points. $\mathbf{r}(\mathbf{z})$ is a weighted linear combination of a prespecifed set of scalar basis functions $\{\phi_k(\mathbf{z}) \mid k = 1 \ldots p\}$. 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 $\check{\mathbf{x}}$, $\mathbf{x} = \mathbf{r}(\check{\mathbf{x}}, \mathbf{z})$ (Section 5). 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 $\mathbf{r}(\check{\mathbf{x}}, \mathbf{z})$, whose basis functions now take the form $\{\phi_k(\check{\mathbf{x}}, \mathbf{z}) \mid k = 1 \ldots p\}$.

Our solutions are well-regularized in the sense that the weight vectors $\mathbf{a}_k$ are damped to control overfitting 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. For a linear basis ($\phi_k(\mathbf{z}) = k\text{th}$ component of $\mathbf{z}$), the sparse solution obtained by the RVM allows the system to select relevant input *features* (components). For a kernel basis—$\phi_k(\mathbf{z}) \equiv K(\mathbf{z}, \mathbf{z}_k)$ for some kernel function $K(\mathbf{z}, \mathbf{z}')$ and centers $\mathbf{z}_k$—relevant training *examples* are selected, allowing us to prune a large training data set and retain only a minimal subset.

1.3 Organization

Section 2 describes our image descriptors and body pose representation. Section 3 gives an outline of our regression methods. Section 4 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. Section 5 describes a tracking based regression framework capable of resolving these ambiguities, with the formulation of a fully regressive tracker in Section 5.1 and an alternative Condensation-based tracker in Section 5.2. Finally, Section 6 concludes with some discussions and directions for future work.

2 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.

2.1 Images as Shape Descriptors

2.1.1 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: 1) they can be extracted moderately reliably from images, at least when robust background or motion-based segmentation is available and problems with shadows are avoided, 2) they are

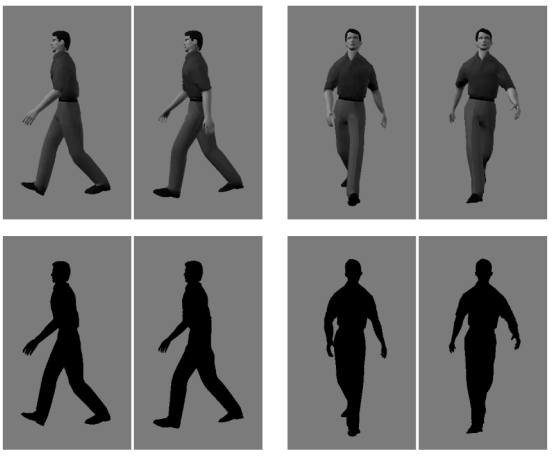


Fig. 1. Different 3D poses can have very similar image observations, causing the regression from image silhouettes to 3D pose to be inherently multivalued. The legs and the arms are reversed in the first two images, for example.

insensitive to irrelevant surface attributes like clothing color and texture, and 3) they encode a great deal of useful information about 3D pose without the need of any labeling information.[1]

Two factors limit the performance attainable from silhouettes: 1) 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*. 2) Silhouettes leave 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 the exact poses of arms or hands that fall within (are "occluded" by) the torso's silhouette are. 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.

2.1.2 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 [16], [6], so we begin by computing local descriptors at regularly spaced points on the edge of the silhouette. About 400-500 points are used, which corresponds to a one pixel spacing on silhouettes of size 64×128 pixels such as those in our training set. We use shape contexts (histograms of local edge pixels into log-polar bins [6]) to encode the local silhouette shape at a range of scales quasilocally, over regions of diameter similar to the length of a limb. The scale of the shape contexts is calculated as a function of the overall silhouette size, making the representation invariant to the overall scale of a silhouette. See Fig. 2c. 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]. Our shape contexts contain 12 angular $\times$ five radial bins, giving rise to 60-dimensional histograms. The silhouette shape is thus encoded as a 60D distribution (in fact, as a noisy multibranched curve, but we treat it as a distribution) in the shape context space.

Matching silhouettes is therefore reduced to matching distributions in shape context space. To implement this, a second level of histogramming is performed: We vector quantize the shape context space and use this to reduce the distribution of each silhouette to a 100D histogram. Silhouette comparison is thus finally reduced to a comparison of 100D histograms. The 100 center 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. 3. Other center selection methods give similar results. For a given silhouette, a 100D histogram $\mathbf{z}$ is built by allowing each of its 60D context vectors to vote softly into the few center-classes nearest to it and accumulating the scores of all of the silhouette's context vectors. The votes are computed by placing a Gaussian at each center and computing the posterior probability for each shape context to belong to each center/bin. We empirically set the common variance of the Gaussians such that each shape context has significant votes into four to five centers. 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 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. 4).

2.2 Body Pose as Joint Angles

We recover 3D body pose (including orientation with respect to the camera) as a real 55D vector $\mathbf{x}$, including three joint angles for each of the 18 major body joints shown in Fig. 2f. 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 $3 \times 18 + 1 = 55$ parameters.

We stress that our framework is inherently "model-free" and is independent of the choice of this pose representation. The method has also been tested on a different parameterization of the body joint angles as a 44D vector. The system itself has no explicit body model or rendering model 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 in the form

1. We believe that any representation (Fourier coefficients, etc.) based on treating the silhouette shape as a continuous parametrized curve is inappropriate 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 with respect to shape.

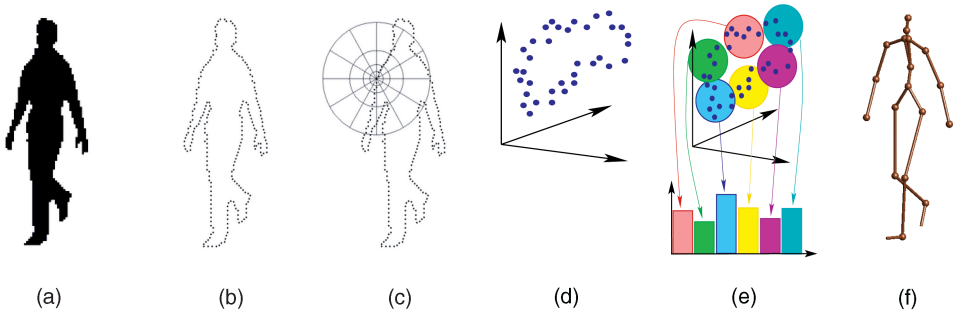


Fig. 2. A step by step illustration of our method: (a) Input silhouette extracted using background subtraction. (b) Sampled edge points. (c) Local shape contexts computed on edge points. (d) Distribution in shape context space. (e) *Soft* vector quantization to obtain a single histogram. (f) Three-dimensional pose obtained by regressing on this histogram.

of Euler angles. Our regression method handles such redundant output representations without problems.

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

3 Regression Methods

This section describes the regression methods that we have evaluated for recovering 3D human body pose from the above image descriptors. We represent the output pose by real vectors $\mathbf{x} \in \mathbb{R}^m$ and the input shape as vectors $\mathbf{z} \in \mathbb{R}^d$.

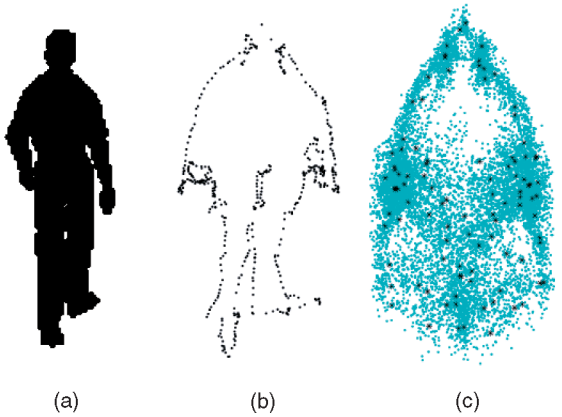


Fig. 3. Silhouette encoding using local shape context descriptors. Silhouette shape (a) is encoded as a fuzzy form in the shape context space (b). The figure shows a projection on the first two principal components of the distribution of 60D shape context vectors computed on the edge points of this silhouette. (c) The projection of all context vectors from a training data sequence. 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. Also shown here are k-means centers that are used to vector quantize each silhouette's distribution into a single histogram. See text.

We assume that the relationship between $\mathbf{z}$ and $\mathbf{x}$—which a priori, given the ambiguities of pose recovery, might be multivalued and, hence, relational rather than functional—can be approximated functionally as a linear combination of a prespecified set of basis functions:

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

Here, $\{\phi_k(\mathbf{z}) \mid k = 1 \ldots p\}$ 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 $\mathbf{A} \equiv \begin{pmatrix}\mathbf{a}_1 & \mathbf{a}_2 \cdots \mathbf{a}_p\end{pmatrix}$ and the basis functions into a $\mathbb{R}^p$-valued function $\mathbf{f}(\mathbf{z}) = \begin{pmatrix}\phi_1(\mathbf{z}) & \phi_2(\mathbf{z}) \cdots \phi_p(\mathbf{z})\end{pmatrix}^\top$. To allow for a constant offset $\mathbf{x} = \mathbf{A}\mathbf{f} + \mathbf{b}$, we can include $\phi(\mathbf{z}) \equiv 1$ in $\mathbf{f}$.

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

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

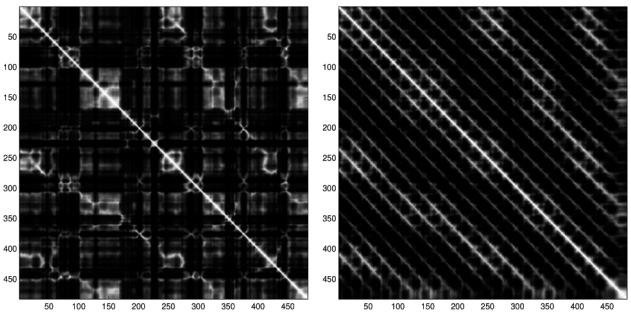


Fig. 4. Pairwise similarity matrices for (a) image silhouette descriptors and (b) true 3D poses for a 483-frame sequence of a person walking in a decreasing spiral. The light offdiagonal bands that are visible in both matrices denote regions of comparative similarity linking corresponding poses on different cycles of the spiral. This indicates that our silhouette descriptors do indeed capture a significant amount of pose information. (The light antidiagonal ripples in the 3D pose matrix arise because the standing-like poses at the middle of each stride have midrange joint values and, hence, are closer on average to other poses than the "stepping" ones at the end of strides).

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

$$\mathbf{A} \;:=\; \arg\min_{\mathbf{A}} \Big\{ \;\|\mathbf{A}\,\mathbf{F} - \mathbf{X}\|^2 + R(\mathbf{A}) \;\Big\}, \tag{3}$$

where $\|.\|$ denotes the Frobenius norm. Note that the dependence on $\{\phi_k(-)\}$ and $\{\mathbf{z}_i\}$ is encoded entirely in the numerical matrix $\mathbf{F}$.

3.1 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 *damped least squares* or *ridge* regressor, which minimizes

$$\|\mathbf{A}\,\tilde{\mathbf{F}} - \tilde{\mathbf{X}}\|^2 \;\;:=\;\; \|\mathbf{A}\,\mathbf{F} - \mathbf{X}\|^2 + \lambda\,\|\mathbf{A}\|^2, \tag{4}$$

where $\tilde{\mathbf{F}} \equiv (\mathbf{F} \;\; \lambda\,\mathbf{I})$ and $\tilde{\mathbf{X}} \equiv (\mathbf{X} \;\; \mathbf{0})$. The solution can be obtained by solving the linear system $\mathbf{A}\,\tilde{\mathbf{F}} = \tilde{\mathbf{X}}$ (i.e., $\tilde{\mathbf{F}}^\top\,\mathbf{A}^\top = \tilde{\mathbf{X}}^\top$) for $\mathbf{A}$ in least squares,[2] using QR decomposition or the normal equations. Ridge solutions are not equivariant under relative scaling of input dimensions, so we usually scale the inputs 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 over-damping (forcing $\mathbf{A}$ toward $\mathbf{0}$ so that the regressor systematically underestimates the solution).

3.2 Relevance Vector Regression

Relevance Vector Machines (RVMs) [28], [29] 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 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 $R(a) = \nu \log \|a\|$. This has an effect of pushing unnecessary parameters to zero, thus producing a sparse model, i.e., the RVM automatically selects the most "relevant" basis functions to describe the problem. The functional form that we minimize is given by

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

where $\mathbf{a}_k$ are the columns of $\mathbf{A}$. Details of the minimization algorithm and a discussion of the sparseness properties of the RVM are given in the Appendix.

3.3 Choice of Basis

We tested two kinds of regression bases $\mathbf{f}(\mathbf{z})$: 1) *Linear bases*, $\mathbf{f}(\mathbf{z}) \equiv \mathbf{z}$, simply return the input vector, so the regressor is linear in $\mathbf{z}$ and the RVM selects relevant *features* (components of $\mathbf{z}$). 2) *Kernel bases*, $\mathbf{f}(\mathbf{z}) = (K(\mathbf{z},\mathbf{z}_1) \cdots K(\mathbf{z},\mathbf{z}_n))^\top$, are based on a kernel function $K(\mathbf{z},\mathbf{z}_i)$ instantiated at training examples $\mathbf{z}_i$, so the RVM effectively selects relevant *examples*. Our experiments show that linear bases on our already highly nonlinear features work well, but that kernelization gives a small but useful improvement—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 $K(\mathbf{z},\mathbf{z}_i) = e^{-\beta\|\mathbf{z}-\mathbf{z}_i\|^2}$ with β estimated from the scatter matrix of the training data, but other β values within a factor of two from this value give very similar results.

4 Pose from Static Images

We conducted experiments using a database of motion capture data for a 54 degrees of freedom body model (three angles for each of 18 joints, including body orientation with respect to the camera). We report mean (over all 54 angles) RMS absolute difference errors between the true and estimated joint angle vectors, in degrees:

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

where $x \bmod \pm 180^\circ \;\equiv\; (x + 180^\circ) \bmod 360^\circ - 180^\circ$ reduces angles to the interval $[-180^\circ, +180^\circ]$. The training silhouettes were created by using POSER to render the motion captured poses.

We compare the results of regressing body pose $\mathbf{x}$ (in the 55D representation of Section 2.2) against silhouette descriptors $\mathbf{z}$ (the 100D histograms of Section 2.1) using ridge, RVM, and SVM regression methods on linear and kernel bases. Ridge and RVM regression use quadratic loss functions to measure $\mathbf{x}$-space prediction errors, as described in Section 3, while SVM regression [31], [25] uses the ϵ-insensitive loss function and a linear programming method for training. The results shown here use SVM-*Light* [14] for training.

4.1 Implicit Feature Selection

Linear RVM regression reveals which of the original input features encode useful pose information, as the RVM directly selects relevant components of $\mathbf{z}$. 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 and neck, the two arms, and the two legs—and trained separate linear RVM regressors for each subset. Fig. 5 shows that similar validation-set errors are attained for each part,

2. If a constant offset $\mathbf{x} = \mathbf{A}\mathbf{f} + \mathbf{b}$ is included, $\mathbf{b}$ must not be damped, so the system takes the form $(\mathbf{A} \;\; \mathbf{b})\;\tilde{\mathbf{F}} = \tilde{\mathbf{X}}$, where $\tilde{\mathbf{X}} \equiv (\mathbf{X} \;\; \mathbf{0})$ and

$$\tilde{\mathbf{F}} \equiv \begin{pmatrix} \mathbf{F} & \lambda\,\mathbf{I} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}.$$

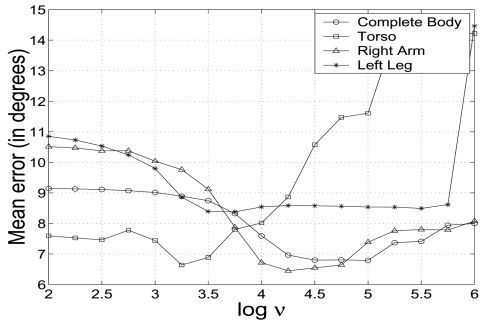


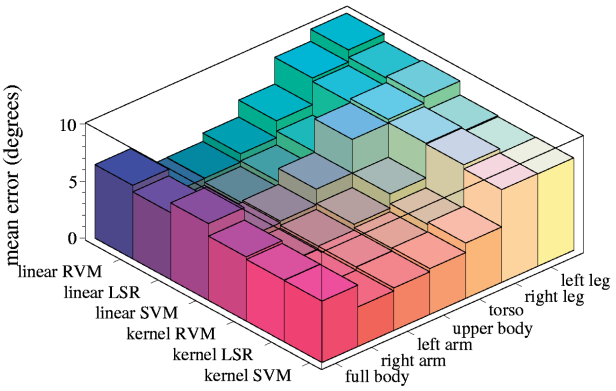
Fig. 5. The mean test-set fitting error for various body parts, versus the linear RVM sparseness parameter ν. The minima indicate the optimal sparsity/regularization settings for each part. Limb regressors are sparser than body or torso ones. The whole body regressor retains 23 features; the torso, 31; the right arm, 10; and the left leg, 7.

but the optimal regularization level is significantly smaller (there is less sparsity) for the torso than for the other parts. Fig. 6 shows the silhouette points whose contexts contribute to the features (histogram classes) that were selected as relevant, for several parts and poses. The main observations are that the regressors are indeed sparse—only about 10 percent of the 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 nonlocality 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.

4.2 Performance Analysis

Fig. 7 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 of it—at optimal regularizer settings computed using 2-fold cross validation. All output parameters are

Fig. 6. The 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 and neck parameter values over different view points and poses. On average, about 10 features covering about 10 percent of the silhouette suffice to estimate the pose of each body part.



(a)

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

(b)

Fig. 7. (a) A summary of our various regressors' performance on different combinations of body parts for the spiral walking test sequence. (b) Error measures for the full body using Gaussian kernel bases with the corresponding number of support vectors retained.

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 nonlinear) 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 percent) of the 2,636 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 and plan to investigate an ϵ-insensitive loss RVM to verify this. The overall similarity of the results obtained from the three different regressors confirms that our representation and framework are insensitive to the exact method of regression used.

Fig. 8 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°. The RMS errors for individual body angles depend on the observability and on the ranges of variation of these angles (in parentheses): body heading angle, $17^\circ(360^\circ)$; left shoulder angle, $7.5^\circ(51^\circ)$; and right hip angle, $4.2^\circ(47^\circ)$. Fig. 9a 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 poses where the silhouette is ambiguous and might easily arise from any of several possible poses.

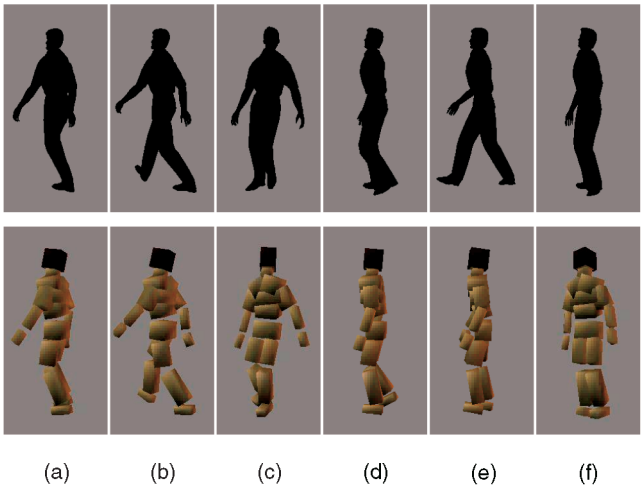


Fig. 8. Some sample pose reconstructions for a spiral walking sequence not included in the training data. The reconstructions were computed with a Gaussian kernel RVM, using only 156 of the 2,636 training examples. The mean angular error per degree of freedom over the whole sequence is 6.0°. While (a)-(c) show accurate reconstructions, (d)-(f) are examples of misestimation: (d) illustrates a label confusion (the left and right legs have been interchanged), (e) and (f) are examples of compromised solutions where the regressor has averaged between two or more distinct possibilities. Using single images alone, we find ~ 15 percent of our results are misestimated.

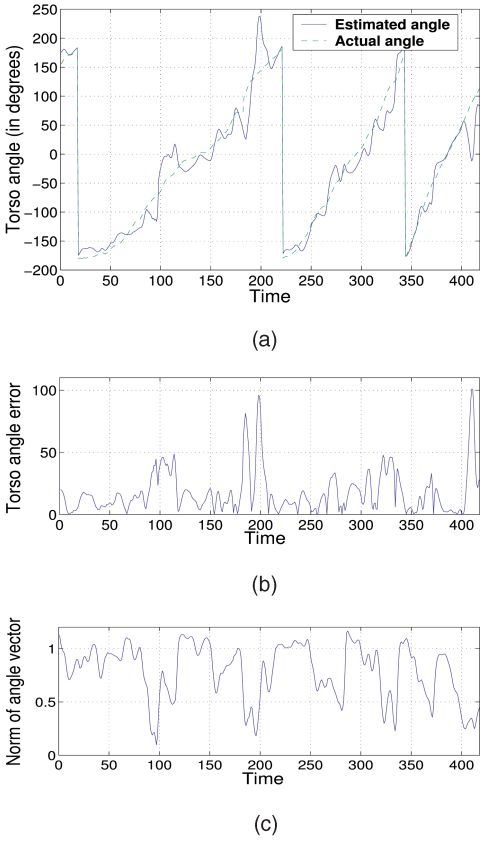


Fig. 9. (a) The estimated body heading (azimuth θ) over 418 frames of the spiral walking test sequence, compared with its actual value from motion capture. (b) and (c) Episodes of high estimation error are strongly correlated with periods when the norm of the $(\cos\theta, \sin\theta)$ vector that was regressed to estimate θ becomes small. These occur when similar silhouettes arise from very different poses, so that the regressor is forced into outputting a compromise solution.

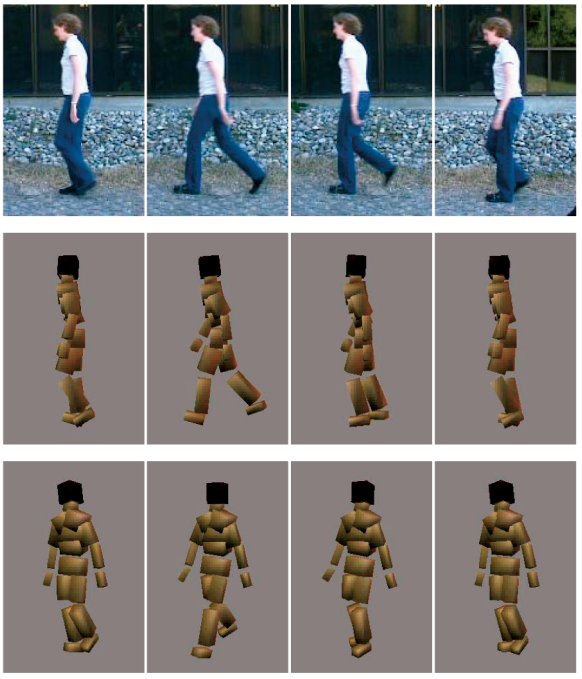


Fig. 10. Three-dimensional poses reconstructed from some real test images using a single image for each reconstruction (the images are part of a sequence from www.nada.kth.se/~hedvig/data.html). The middle and lower rows respectively show the estimates from the original viewpoint and from a new one. 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.

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. 9.

Fig. 10 shows reconstruction results on some real images. A relatively unsophisticated background subtraction method was used to extract the silhouettes, but this demonstrates the method's robustness to imperfect visual features. The last example illustrates the problem of silhouette ambiguity: the method returns a pose with the left knee bent instead of the right one because the silhouette looks the same in the two cases, causing a glitch in the output pose.

Although our results are already very competitive with others presented in the literature, 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), or 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 compromise between two different solutions. 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 edges are often invisible and useful ones have to be distinguished from irrelevant clothing texture. Furthermore, even without these limb labeling ambiguities, depth related ambiguities exist and remain an issue. By keying on experimentally observed poses, our single image method already reduces this ambiguity significantly, but the subtle cues that human beings rely on to disambiguate multiple solutions remain inaccessible. The following section describes how we exploit temporal continuity within our regression framework to reduce this ambiguity.

5 Tracking and Regression

This section describes a novel "discriminative" tracking framework that reconstructs the most likely 3D pose at each time step by fusing pose predictions from a learned dynamical model into our single image regression framework. The 3D pose can only be observed indirectly via ambiguous and noisy image measurements, so we start by considering the Bayesian tracking framework which represents our knowledge about the state (pose) $\mathbf{x}_t$ given the observations up to time t as 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 usually 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, 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 backward 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 modeling appearance details that are irrelevant for predicting pose. Just as we preferred to learn a "diagnostic" regressor $\mathbf{x} = \mathbf{x}(\mathbf{z})$, not a generative predictor $\mathbf{z} = \mathbf{z}(\mathbf{x})$ for pose reconstruction, we prefer to learn a diagnostic model $p(\mathbf{x}|\mathbf{z})$ for the pose $\mathbf{x}$ given the observations $\mathbf{z}$—cf. the difference between generative and discriminative classifiers and the regression-based trackers of [15], [32]. However, 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 labeling. 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 multivalued. 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.

One approach to this is to learn a multivalued representation, and we are currently developing a method of this type. Here, we take another approach, reducing the ambiguity by working incrementally from the previous few states[3] $\mathbf{x}_{t-1}, \ldots$ (cf. [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 possible observation-based estimates $\mathbf{x}(\mathbf{z})$ lies near $\check{\mathbf{x}}$. 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 local and, hence, nonlinear in $\check{\mathbf{x}}_t$. 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 $\mathbf{x}_t$.

5.1 Learning the Regression Models

Our discriminative tracking framework now has two levels of regression:

5.1.1 Dynamical (Prediction) Model

Human body dynamics can be modeled 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 (cf. [3]). To ensure dynamical stability and avoid overfitting, we learn the autoregression for $\check{\mathbf{x}}_t$ in the following form:

$$\check{\mathbf{x}}_t \;\equiv\; (\mathbf{I} + \mathbf{A})(2\mathbf{x}_{t-1} - \mathbf{x}_{t-2}) + \mathbf{B}\, \mathbf{x}_{t-1}, \tag{7}$$

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.

5.1.2 Likelihood (Correction) Model

Now, consider the observation model. As discussed above, the underlying density $p(\mathbf{x}_t \mid \mathbf{z}_t)$ 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 can be removed by feeding $\check{\mathbf{x}}_t$ into the regression model. A combined regressor could be formulated in several ways. Linearly combining $\check{\mathbf{x}}_t$ with the estimate $\mathbf{x}_t$ from (1) would only smooth the results, reducing jitter while still continuing to give wrong solutions when (1) returns a wrong estimate. Instead, we build a state sensitive observation update by including a nonlinear dependence on $\check{\mathbf{x}}_t$ with $\mathbf{z}_t$ in the observation-based regressor. Our full regression model also

3. The ambiguities persist for several frames so regressing the pose $\mathbf{x}_t$ against a sequence of the last few silhouettes $(\mathbf{z}_t, \mathbf{z}_{t-1}, \ldots)$ does not suffice.

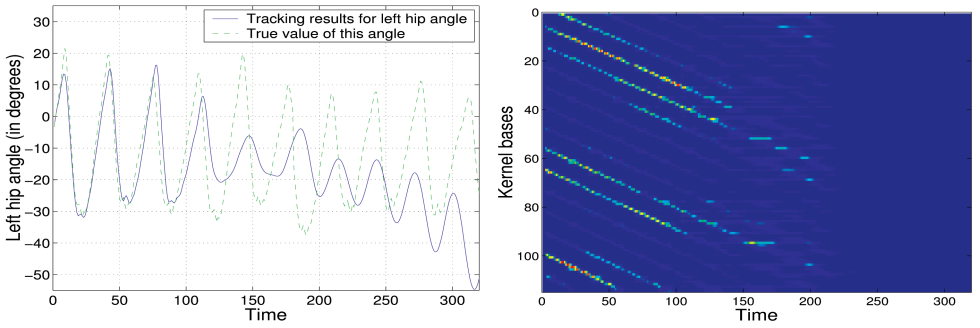


Fig. 11. An example of mistracking caused by an over-narrow pose kernel K_x. The kernel width is set to 1/10 of the optimal value, causing the tracker to lose track from about $t = 120$, after which the state estimate drifts away from the training region and all kernels stop firing by about $t = 200$. (a) The variation of a left hip angle parameter for a test sequence of a person walking in a spiral. (b) 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 eight steps. The similarity between adjacent steps and between different circuits is clearly visible, showing that the regressor can locally still generalize well.

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

$$\hat{\mathbf{x}}_t \;=\; \mathbf{C}\,\check{\mathbf{x}}_t + \sum_{k=1}^{p} \mathbf{d}_k\,\phi_k(\check{\mathbf{x}}_t, \mathbf{z}_t) \;\equiv\; \begin{pmatrix}\mathbf{C} & \mathbf{D}\end{pmatrix}\begin{pmatrix}\check{\mathbf{x}}_t \\ \mathbf{f}(\check{\mathbf{x}}_t, \mathbf{z}_t)\end{pmatrix}. \tag{8}$$

Here, $\{\phi_k(\mathbf{x}, \mathbf{z}) \mid k = 1 \ldots p\}$ is a set of scalar-valued nonlinear 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 $\mathbf{f}(\mathbf{x}, \mathbf{z}) \equiv \big(\phi_1(\mathbf{x}, \mathbf{z}), \ldots, \phi_p(\mathbf{x}, \mathbf{z})\big)^\top$ and an $m \times p$ weight matrix $\mathbf{D} \equiv \big(\mathbf{d}_1, \ldots, \mathbf{d}_p\big)$. In the experiments reported here, we used instantiated-kernel bases of the form

$$\phi_k(\mathbf{x}, \mathbf{z}) = K_x(\mathbf{x}, \mathbf{x}_k) \cdot K_z(\mathbf{z}, \mathbf{z}_k), \tag{9}$$

where $(\mathbf{x}_k, \mathbf{z}_k)$ is a training example and K_x, K_z are 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 $K_z(\mathbf{z}, \mathbf{z}_k) = e^{-\beta_z \|\mathbf{z}-\mathbf{z}_k\|^2}$. Using Gaussians in joint $(\mathbf{x}, \mathbf{z})$ space makes examples relevant only if they have similar image silhouettes *and* similar underlying poses to training examples.

5.1.3 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. 11 illustrates this effect for an $\mathbf{x}$-kernel a factor of 10 narrower than the optimum. The method is thus somewhat sensitive to the kernel width parameters, but, after fixing good values by cross-validation on an independent sequence, we observed accurate performance over a range of about two on β_x and about four on β_z.

5.1.4 Neutral versus Damped Dynamics

The coefficient matrix $\mathbf{C}$ in (8) 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). At the other extreme, $\mathbf{C} \equiv \mathbf{0}$ gives largely observation-based state estimates with little dependence on the dynamics. An intermediate setting with $\mathbf{C}$ near $\mathbf{I}$ 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. 12). This simple heuristic setting gives very similar results to the model obtained by learning the full matrix $\mathbf{C}$.

5.2 A Condensation-Based Viewpoint

The discriminative/regressive approach can be integrated into a Condensation [13] style Bayesian tracking framework.

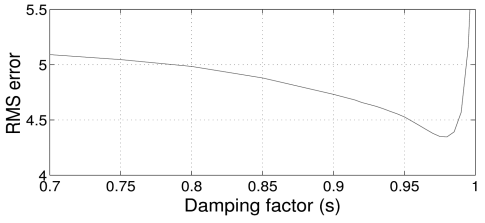


Fig. 12. The variation of the RMS test-set tracking error with damping factor s. See the text for discussion.

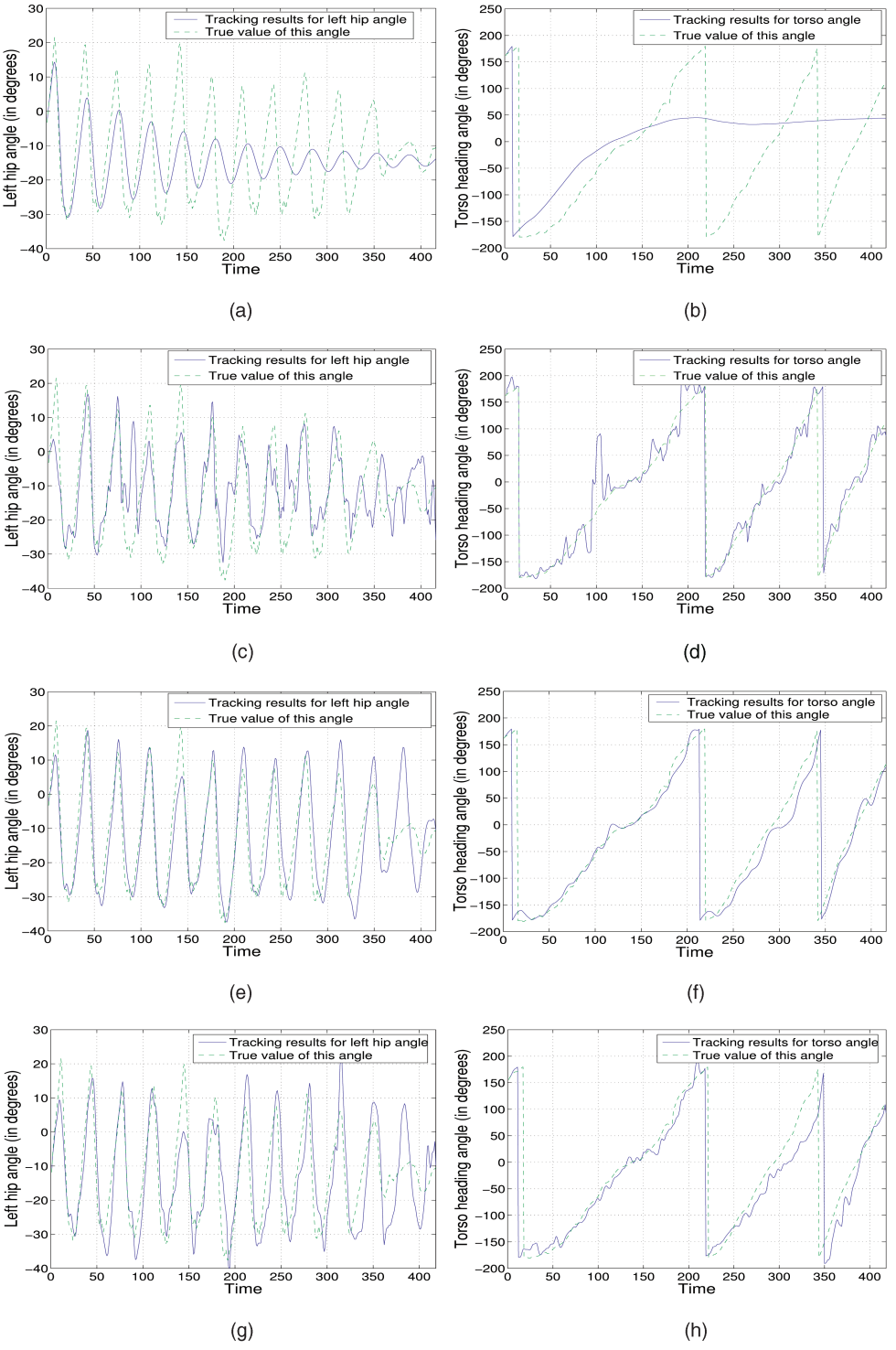


Fig. 13. Sample tracking results on a spiral walking test sequence. (a) and (b) Variation of the left hip-angle and overall body rotation parameters, as predicted by a pure dynamical model initialized at $t = \{0, 1\}$. (c) and (d) Estimated values of these angles from regression on observations alone (i.e., no initialization or temporal information). (e) and (f) Results from our novel joint regressor, obtained by combining dynamical and state + observation-based regression models. (g) and (h) Condensation-based tracking, showing a smoothed trajectory of the most likely particle at each time step. Note that the overall body rotation angle wraps around at 360°, i.e., $\theta \simeq \theta \pm 360^\circ$.

Assuming the state information from the current observation is independent of state information from dynamics, we obtain

$$p(\mathbf{x}_t \mid \mathbf{z}_t, \mathbf{x}_{t-1}, \ldots) \propto \frac{p(\mathbf{z}_t \mid \mathbf{x}_t)}{p(\mathbf{z}_t)}\, p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \ldots). \tag{10}$$

A dynamical model gives us $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \ldots)$. We must now fuse in the information from $\mathbf{z}_t$. The way to do this is to multiply by the contrast $\frac{p(\mathbf{x}_t, \mathbf{z}_t)}{p(\mathbf{x}_t)p(\mathbf{z}_t)} = \frac{p(\mathbf{x}_t \mid \mathbf{z}_t)}{p(\mathbf{x}_t)} = \frac{p(\mathbf{z}_t \mid \mathbf{x}_t)}{p(\mathbf{z}_t)}$. Here, $p(\mathbf{x}_t)$ or $p(\mathbf{z}_t)$ are vague priors assuming no knowledge of the previous state, so they have little influence. Often, the contrast term is approximated by the likelihood $p(\mathbf{z}_t \mid \mathbf{x}_t)$ by

Fig. 14. Some sample pose reconstructions for the spiral walking sequence using the tracking method. This sequence (not included in the training data) corresponds to Figs. 13e and 13f. The reconstructions were computed with a Gaussian kernel RVM which retained only 18 percent of the training examples. The average RMS estimation error per degrees of freedom over the whole sequence is 4.1°.

building a generative model for image observations. In our discriminative model, we ignore the dependence on $p(\mathbf{x})$ and estimate a noise model for the regressor to directly model $p(\mathbf{x}_t \mid \mathbf{z}_t)$ as a Gaussian centered at $\mathbf{r}(\check{\mathbf{x}}_t, \mathbf{z}_t)$. The term $\sum_{k=1}^{p} \mathbf{d}_k \, \phi_k(\check{\mathbf{x}}_t, \mathbf{z}_t)$ in (8) is thought of as parameterizing the observation-based state density that replaces the likelihood term. Thus, the dynamical model from Section 5.1.1 is used to generate an estimate of the 3D pose distribution $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \ldots)$ and samples $(\check{\mathbf{x}}_t^i)$ from this distribution are then assigned weights using a Gaussian model centered at the regressor output $\sum_{k=1}^{p} \mathbf{d}_k \, \phi_k(\check{\mathbf{x}}_t, \mathbf{z}_t)$ with covariance learned from the training data.

5.3 Tracking Results

We trained the new regression model (8) on our motion capture data as in Section 4. For these experiments, we used eight different sequences totaling about 2,000 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 as described in Section 5.1.1. When training the observation model, 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. So, we train the model $\mathbf{x}_t = \mathbf{x}_t(\check{\mathbf{x}}_t, \mathbf{z}_t)$ using a combination of "observed" samples $(\check{\mathbf{x}}_t, \mathbf{z}_t)$ (with $\check{\mathbf{x}}_t$ computed from (7)) and artificial samples that generate $\check{\mathbf{x}}_t$ by Gaussian sampling $\mathcal{N}(\mathbf{x}_t, \Sigma)$ around the training state $\mathbf{x}_t$. The unperturbed 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 $\mathbf{x}_t$ from $\mathbf{z}_t$, using $\check{\mathbf{x}}_t$ only as a hint about the inverse solution to choose. The covariance matrix Σ is chosen to reflect the local scatter of the training example poses, but with increased variance along the tangent to the trajectory at each point so that the model will reliably correct any phase lag between the estimate and true state that builds up during tracking. (Such lags can occur when the observation signal is weak for a few time steps and the model is driven mainly by the dynamical component of the tracker.)

Fig. 13 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. The sequence was not included in the training set, although similar ones were. The purely dynamical model (7) provides good estimates for a few time steps, but gradually damps and drifts out of phase. Such damped oscillations are characteristic of second order autoregressive systems, trained with enough regularization to ensure model stability. The results (from Section 4) based on observations alone without any temporal information are included here again for comparison. These are obtained from (1), which is actually a special case of (8), where $\mathbf{C} = \mathbf{0}$ and $K_x = 1$. Figs. 13e and 13f show that jointly

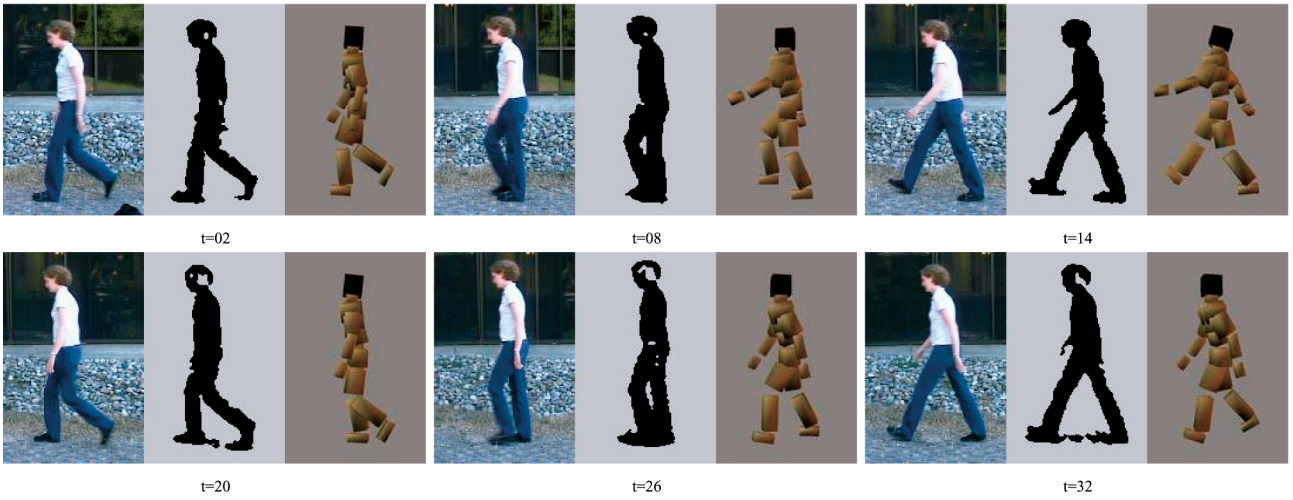


Fig. 15. 3D poses reconstructed from a test video sequence (obtained from www.nada.kth.se/~hedvig/data.html). The presence of shadows and holes in the extracted silhouettes demonstrates the robustness of our shape descriptors—however, a weak or noisy observation signal sometimes causes failure to track accurately, e.g., at $t = 8, 14$, the pose estimates are dominated by the dynamical predictions, which ensure smooth and natural motion but cause slight mistracking of some parameters.

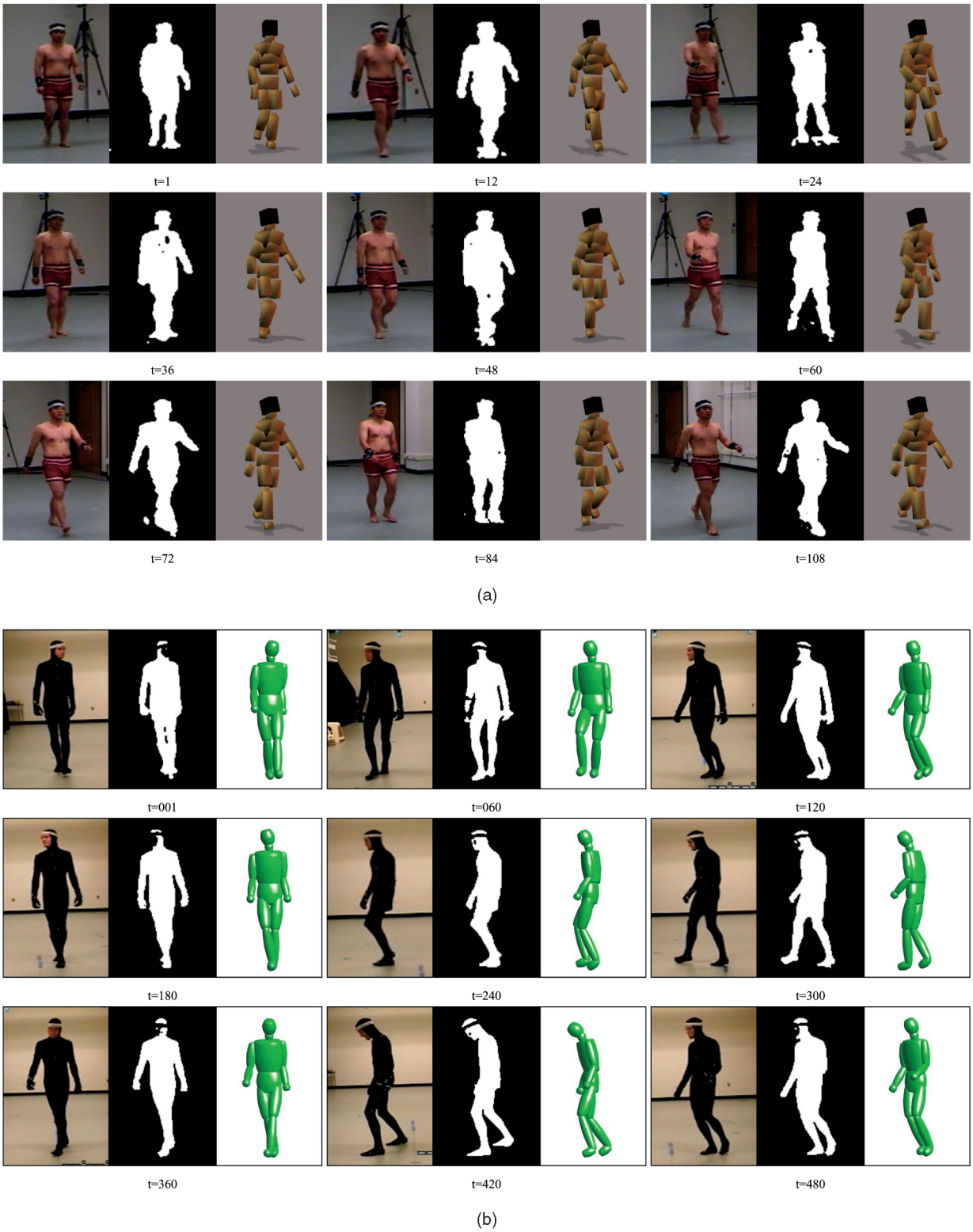


Fig. 16. Three-dimensional pose reconstructions on some example test sequences on which the method was tested. (a) The subject walks toward the camera causing a scale change by a factor of ~ 2. The images and silhouettes have been normalized in scale here for display purposes. (b) The subject often changes heading angle, walking randomly in different directions. The method successfully tracks through 600 frames.

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 Fig. 13e at around t=140 and t=380. At these points, the subject is walking directly toward 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. Results from the Condensation-based tracker described in

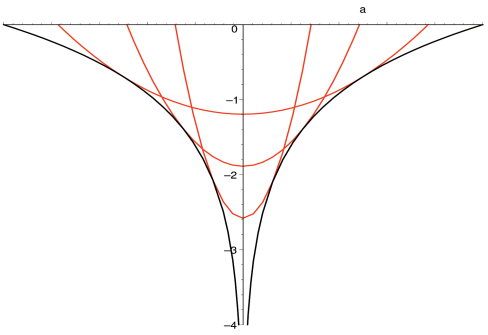


Fig. 17. "Quadratic bridge" approximations to the $\nu \log \|a\|$ regularizers. These are introduced to prevent parameters from prematurely becoming trapped at zero. (See text.)

Section 5.2 are shown in Figs. 13g and 13h. They are very similar to those obtained using the joint regression, but not as smooth.

Fig. 14 shows some silhouettes and the 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°. The Gaussian RVM gives a sparse regressor for (8) involving only 348 (18 percent) of the 1,927 training examples. Well-regularized least squares regression over the same basis gives similar errors, but has much higher storage requirements. Fig. 15 shows reconstruction results on a lateral walking test video sequence. Fig. 16 shows the performance of tracking through different viewpoints. These sequences were obtained from http://mocap.cs.cmu.edu. The first sequence tracks though a scale change by a factor of ~ 2 as the subject walks toward the camera. Note that, as the silhouette representation is invariant to the scale/resolution of an image, no rescaling/downsampling of the test images is required—images and silhouettes in the figure have been normalized in scale only for display purposes. The second sequence is an example of a more complicated motion—the subject often changes heading angle, walking in several different directions. For this example, the system was trained on a somewhat similar sequence of the same person to ensure a wider coverage of his poses. Also, the motion capture data used for training was in a different format, so we used a 44D joint angle representation in this experiment, again emphasizing the methods' independence of the body pose representation.

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 minutes for the RVM with $\sim$2,000 data points and currently about 20 minutes for Shape Context extraction and clustering (this being highly unoptimized Matlab code).

5.3.1 Automatic Initialization

The results shown in Figs. 13 and 14 were obtained by initializing from ground truth, but we also tested the effects of automatic (and, hence, potentially incorrect) initialization. The method is reasonably robust to initialization errors. In an experiment in which the tracker was initialized automatically at each of the time steps using the pure observation model and then tracked forward and backward using the dynamical tracker, the initialization leads to successful tracking in 84 percent of the cases. The failures were the "glitches," where the observation model gave completely incorrect initializations.

6 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 labeled 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. Our kernelized RVM regression trackers retain only about 15-20 percent of the training examples, thus giving a considerable reduction in storage space compared to nearest neighbor methods, which must retain the whole training database.

6.1 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 3D pose space. There are some kinds of motions that the current method cannot handle owing to the use of Euler angles for pose representation. We find that these are susceptible to "gimbal locks," which limits us to tracking (close to) vertical motion and prohibits very complicated motions. As an alternative, we are investigating the use of 3D joint locations in place of angles. We are also working on generalizing the method to deal with partially visible silhouettes and, hence, partial occlusions, and on using linear RVM techniques to identify better (more "relevant") feature sets for human detection, pose recovery, and tracking.

Appendix

The Relevance Vector Machine

Relevance Vector Machines were originally proposed in [28], [29]. They introduce Gaussian priors on each parameter or group of parameters, with each prior being controlled by its own individual scale hyperparameter. The hyperpriors, which obey a power law distribution, can be integrated out analytically to give 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

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. 18. Outline of our RVM training algorithm.

on a. But, the smaller $\|a\|$ becomes, the greater the regularizing force. 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 suffice 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 sparsity-based 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, but this does not prevent RVMs from giving useful results in practice. 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 RVMs, we use 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. 17). The bridging changes the apparent curvature of the cost surfaces, allowing parameters to pass through zero if they need to with less risk of premature trapping. The algorithm is sketched in Fig. 18.

We 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 kth column of $\mathbf{A}$, which selects a common set of relevant basis functions for all components of $\mathbf{y}$. The two 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, we use columnwise priors, i.e., we minimize (5).

Our method is a maximum-a priori type of approach that integrates out the hyperpriors and directly optimizes the parameters $\mathbf{a}$, not a type-II maximum likelihood as the approach described in [29] that integrates out the parameters and optimizes the hyperparameters. See [17] for a discussion of these two philosophies.

ACKNOWLEDGMENTS

This work was supported by the European Union projects VIBES and LAVA and the network of excellence PASCAL.

REFERENCES

[1] A. Agarwal and B. Triggs, "3D Human Pose from Silhouettes by Relevance Vector Regression," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* 2004.
[2] A. Agarwal and B. Triggs, "Learning to Track 3D Human Motion from Silhouettes," *Proc. Int'l Conf. Machine Learning,* 2004.
[3] A. Agarwal and B. Triggs, "Tracking Articulated Motion Using a Mixture of Autoregressive Models," *Proc. European Conf. Computer Vision,* 2004.
[4] V. Athitsos and S. Sclaroff, "Inferring Body Pose without Tracking Body Parts," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* 2000.
[5] V. Athitsos and S. Sclaroff, "Estimating 3D Hand Pose from a Cluttered Image," *Proc. Int'l Conf. Computer Vision,* 2003.
[6] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, Apr. 2002.
[7] C. Bishop, *Neural Networks for Pattern Recognition,* chapter 6. Oxford Univ. Press, 1995.
[8] M. Brand, "Shadow Puppetry," *Proc. Int'l Conf. Computer Vision,* pp. 1237-1244, 1999.
[9] C. Bregler and J. Malik, "Tracking People with Twists and Exponential Maps," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 8-15, 1998.
[10] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning Inverse Kinematics," *Proc. Int'l Conf. Intelligent Robots and Systems,* 2001.
[11] K. Grauman, G. Shakhnarovich, and T. Darrell, "Inferring 3D Structure with a Statistical Image-Based Shape Model," *Proc. Int'l Conf. Computer Vision,* pp. 641-648, 2003.
[12] N. Howe, M. Leventon, and W. Freeman, "Bayesian Reconstruction of 3D Human Motion from Single-Camera Video," *Neural Information Processing Systems,* 1999.
[13] M. Isard and A. Blake, "CONDENSATION—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.
[14] T. Joachims, "Making Large-Scale SVM Learning Practical," *Advances in Kernel Methods—Support Vector Learning.* MIT Press, 1999.
[15] F. Jurie and M. Dhome, "Hyperplane Approximation for Template Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 996-1000, July 2002.
[16] D. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. Int'l Conf. Computer Vision,* pp. 1150-1157, 1999.
[17] D.J. C. MacKay, "Comparison of Approximate Methods for Handling Hyperparameters," *Neural Computation,* vol. 11, no. 5, pp. 1035-1068, 1999.
[18] G. Mori and J. Malik, "Estimating Human Body Configurations Using Shape Context Matching," *Proc. European Conf. Computer Vision,* vol. 3, pp. 666-680, 2002.

[19] D. Ormoneit, H. Sidenbladh, M. Black, and T. Hastie, "Learning and Tracking Cyclic Human Motion," *Neural Information Processing Systems,* pp. 894-900, 2000.
[20] V. Pavlovic, J. Rehg, and J. MacCormick, "Learning Switching Linear Models of Human Motion," *Neural Information Processing Systems,* pp. 981-987, 2000.
[21] Y. Rubner, C. Tomasi, and L.J. Guibas, "A Metric for Distributions with Applications to Image Databases," *Proc. Int'l Conf. Computer Vision,* 1998.
[22] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast Pose Estimation with Parameter Sensitive Hashing," *Proc. Int'l Conf. Computer Vision,* 2003.
[23] H. Sidenbladh, M. Black, and L. Sigal, "Implicit Probabilistic Models of Human Motion for Synthesis and Tracking," *Proc. European Conf. Computer Vision,* vol. 1, 2002.
[24] C. Sminchisescu and B. Triggs, "Kinematic Jump Processes for Monocular 3D Human Tracking," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* June 2003.
[25] A. Smola and B. Schölkpof, "A Tutorial on Support Vector Regression," Technical Report NC2-TR-1998-030, NeuroCOLT2, 1998.
[26] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla, "Filtering Using a Tree-Based Estimator," *Proc. Int'l Conf. Computer Vision,* 2003.
[27] C. Taylor, "Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* 2000.
[28] M. Tipping, "The Relevance Vector Machine," *Neural Information Processing Systems,* 2000.
[29] M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *J. Machine Learning Research,* vol. 1, pp. 211-244, 2001.
[30] K. Toyama and A. Blake, "Probabilistic Tracking in a Metric Space," *Proc. Int'l Conf. Computer Vision,* pp. 50-59, 2001.
[31] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer, 1995.
[32] O. Williams, A. Blake, and R. Cipolla, "A Sparse Probabilistic Learning Algorithm for Real-Time Tracking," *Proc. Int'l Conf. Computer Vision,* 2003.

Ankur Agarwal received the BTech degree in computer science and engineering from the Indian Institute of Technology, Delhi, in 2002 and the MS degree in computer science from the Institut National Polytechnique de Grenoble, France, in 2004. He is currently a PhD student at the GRAVIR laboratory at INRIA Rhône-Alpes in Grenoble. His research interests include machine learning, statistical modeling, and computer vision.

Bill Triggs originally trained as a mathematical physicist at Auckland, Australian National, and Oxford Universities. He has worked extensively on vision geometry (matching constraints, scene reconstruction, autocalibration) and robotics, but his current research focuses on computer vision, pattern recognition, and machine learning for visual object recognition and human motion understanding. He is a Centre National de Recherche Scientifique researcher working in the LEAR (Learning for Vision) team of the GRAVIR laboratory, which is located in INRIA's Rhône-Alpes research unit in Grenoble in the French Alps.